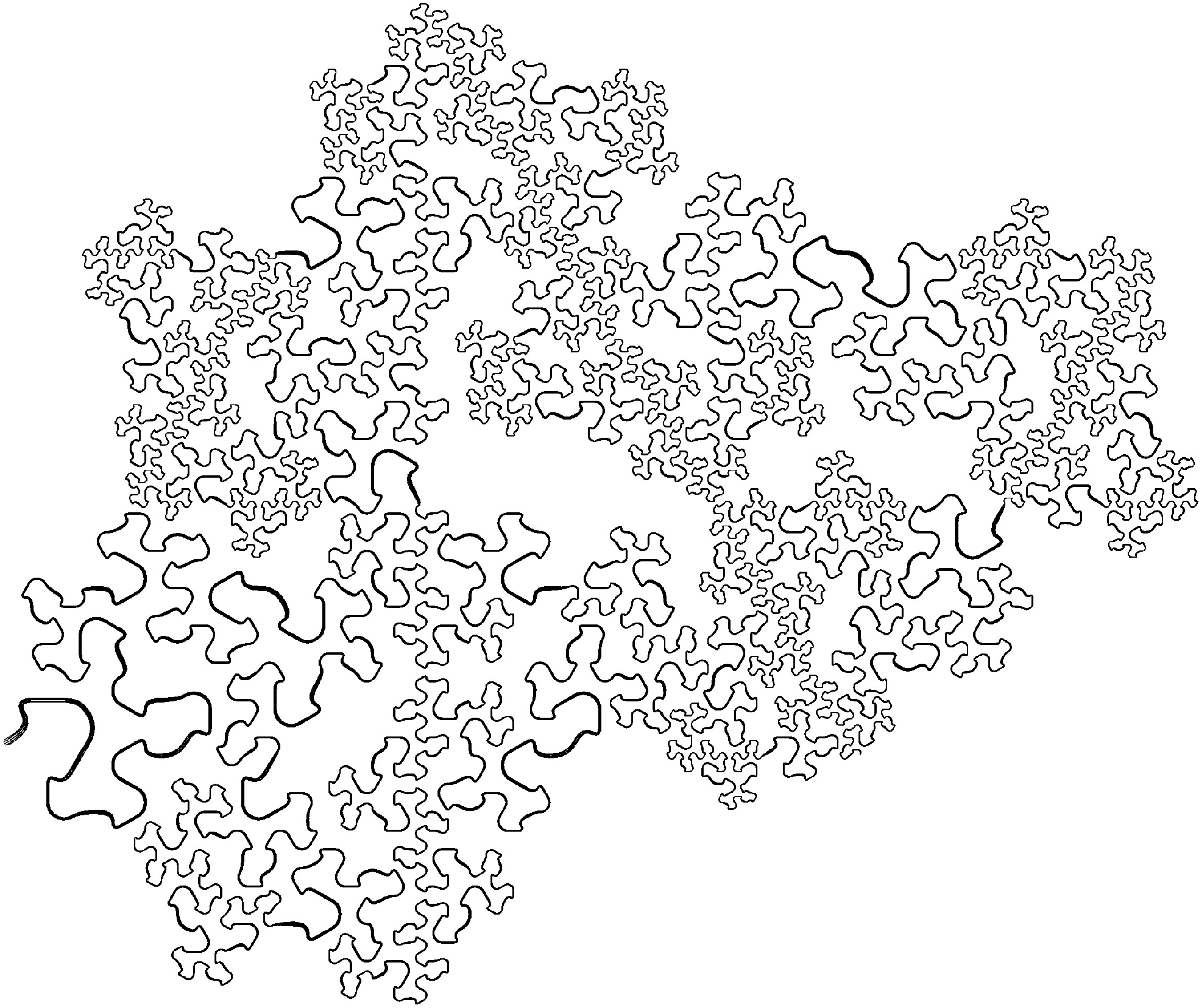
Brain-Filling Curves A Fractal Bestiary

Jeffrey Ventrella



Ventrella, Jeffrey, J.

Brain-filling Curves – A Fractal Bestiary

Second edition

Book web site: www.fractalcurves.com

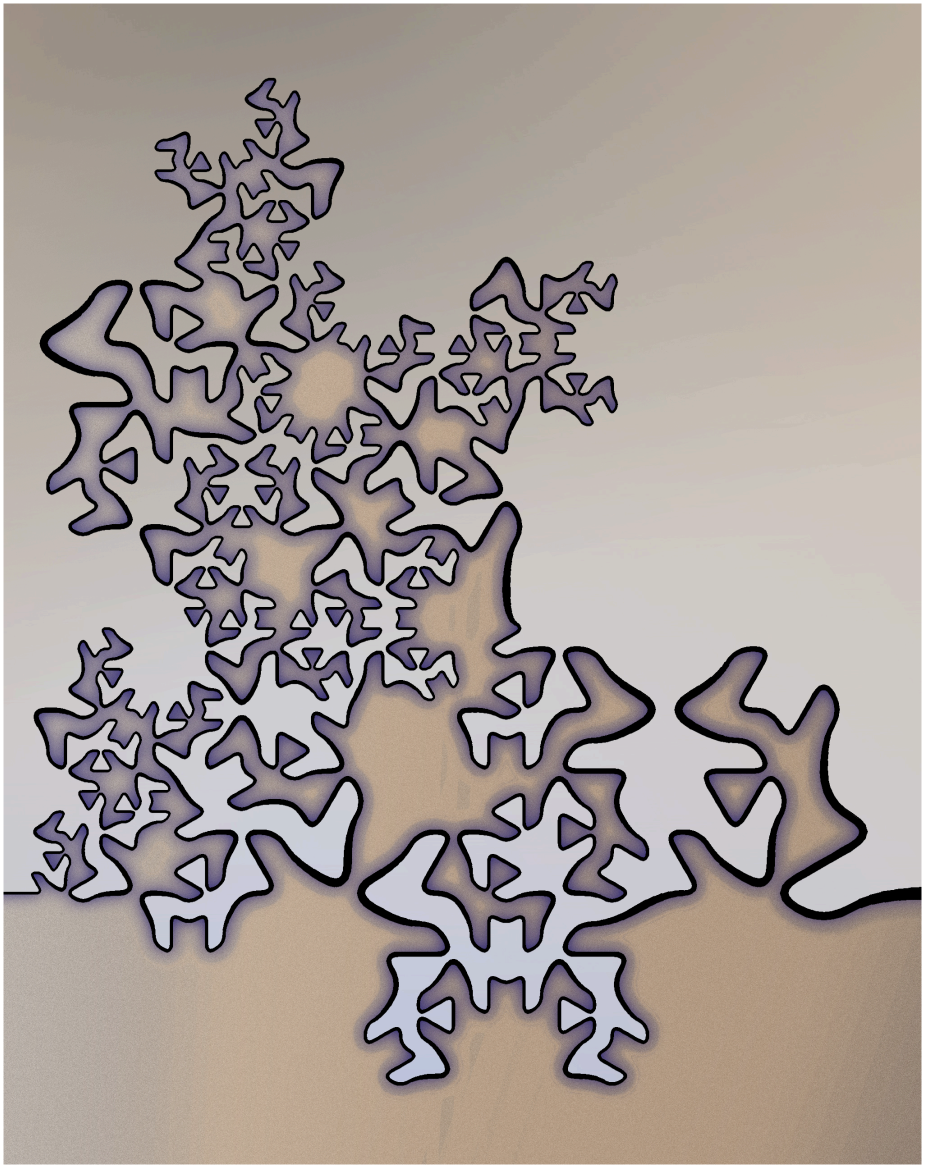
ISBN 978-0-9830546-2-7

Distributed by Lulu.com

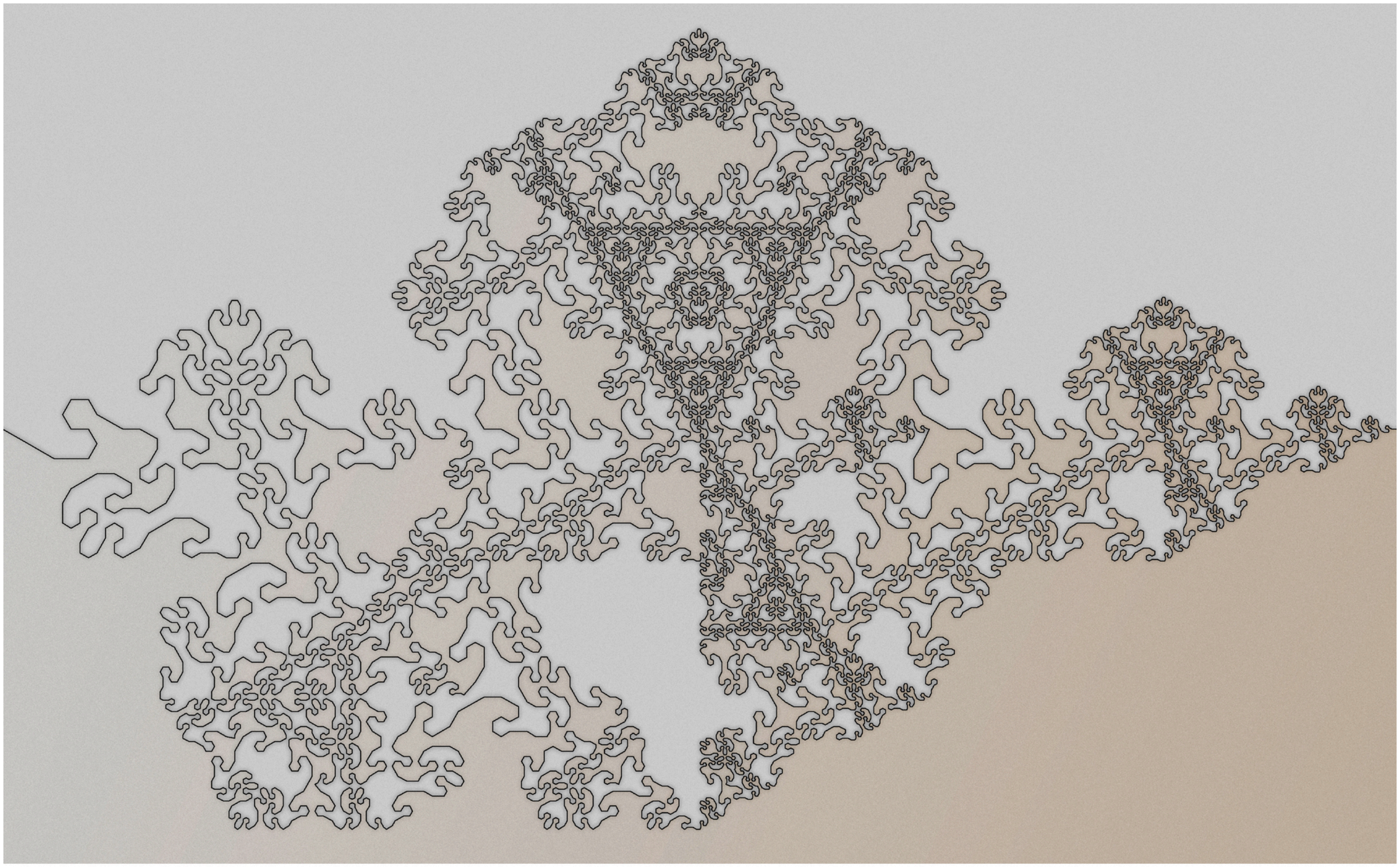
Cover Design by Jeffrey Ventrella

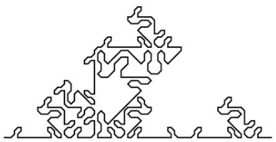
Eyebrain Books

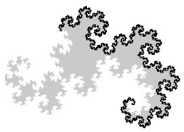
www.eyebrainbooks.com

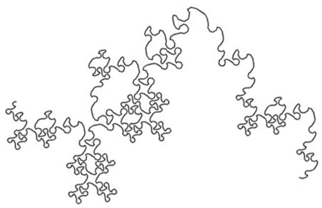


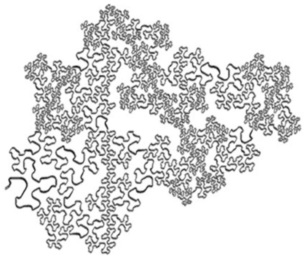
Contents







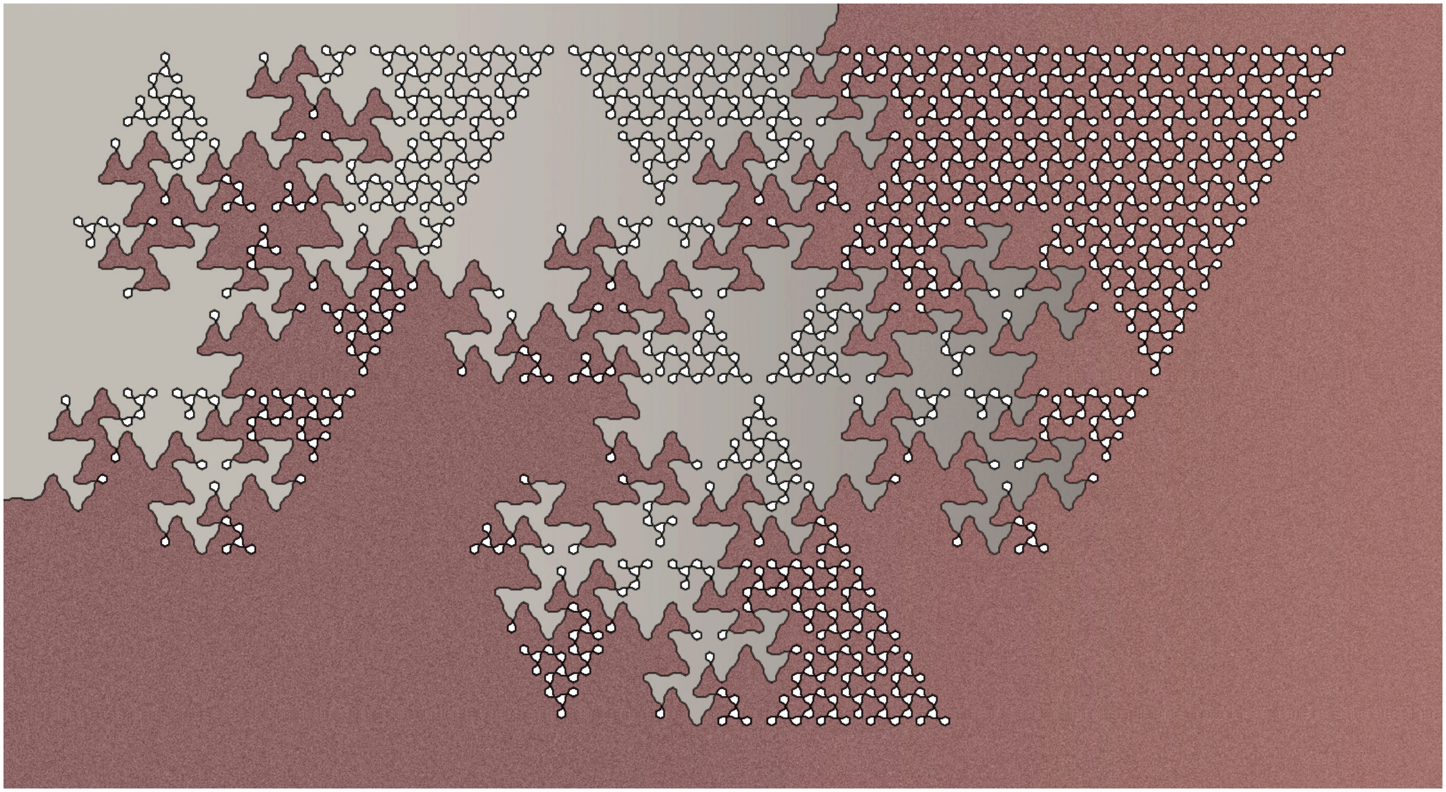




Bes-ti-ar-y *noun (plural bestiaries)* Pronunciation: /ˈbɛstɪəri/
A descriptive or anecdotal treatise on various real or mythical kinds of animals, esp. a medieval work with a moralizing tone. – Oxford Dictionary

Acknowledgements

I want to thank Kirk Israel for reviewing the book and making several keen suggestions. Thanks also to Mirja Holmboe, Gary Walker, Beth O'Sullivan, Richard Montgomery, Portico Bowman, Hiroki Sayama, Eddie Elliot, Karl Stiefvater, Camp Wilson, Claudia L'Amoreaux, and Scott Kim. Thanks to Scott Bowling, brother Philip, and Wayne Bollinger for great fun and programming tips at the very start of this fractal journey. Thanks to professor Ed Zajec, who helped provide an intellectual foundation for making fractal art. Finally, I want to thank Nuala Creed for love and patience with my many obsessions, fractal and otherwise.



1

Horror Vacui

Earth is filled with fractals: shapes and structures that have many levels of detail. Some of these are self-similar: they look the same at multiple scales of magnification. Examples include river basins, which have branch-points on many scales.

Other examples of branching shapes can be found in living organisms: trees, bloodstreams, and bronchial tubes. All of these organic structures demonstrate a need to cover as much space as possible, to maximize tissue surface contact with air or light. Branching forms are wonderful and beautiful, but this book is not strictly about branching forms; it is about very long curly or kinky lines that fill-up areas of a 2D plane. Much like river basins, these long curly lines fully cover an area; they twist and turn so that they can fill-in every corner of space. Interestingly enough: in covering space so thoroughly, these curves acquire branching profiles – of a huge variety, as you will see.

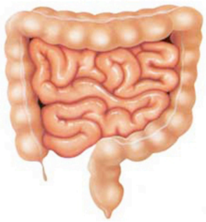
There is a concept in visual art known as: *horror vacui*: “fear of empty space”, illustrated by many art styles throughout history. The specimens in this book have a major case of *horror vacui*: they are truly haters of empty space! A similar concept is found in physics: the notion that “nature abhors a vacuum.” Fractal curves are very rich in metaphorical power – providing models for art, physics, music, and biology. They even evoke philosophical concepts, such as how the paths of our lives are repetitive, and filled with theme and variation – at every time scale.

Our Mascot

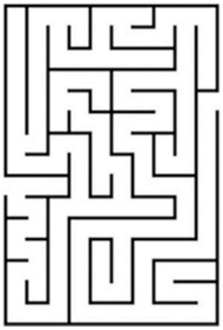
How could I not mention *brain coral* in a book called, "Brain-filling Curves"? Well, I will mention it now! This curious undersea creature – a coral of the family *Faviidae* – is covered with wonderful curly mazes. Hats off to this inspiring creature.

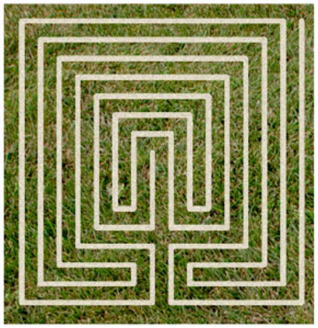
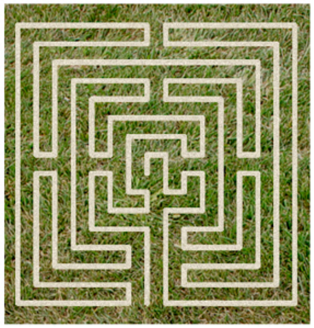
Certain animals – especially really long skinny ones – sometimes find the need to coil themselves up to take up a compact space. We humans also happen to have about seven meters of intestine packed tightly into our abdomens: that's quite a spacefiller!

Any time you have a long flexible linear object, and a desire to have that object take up a small space, you enter into a special realm of geometrical problem-solving – a realm that involves twisting, turning, bending, folding, and wrapping.



This geometric realm is also invoked for the purpose of creating convoluted paths to follow, like a maze or a labyrinth.



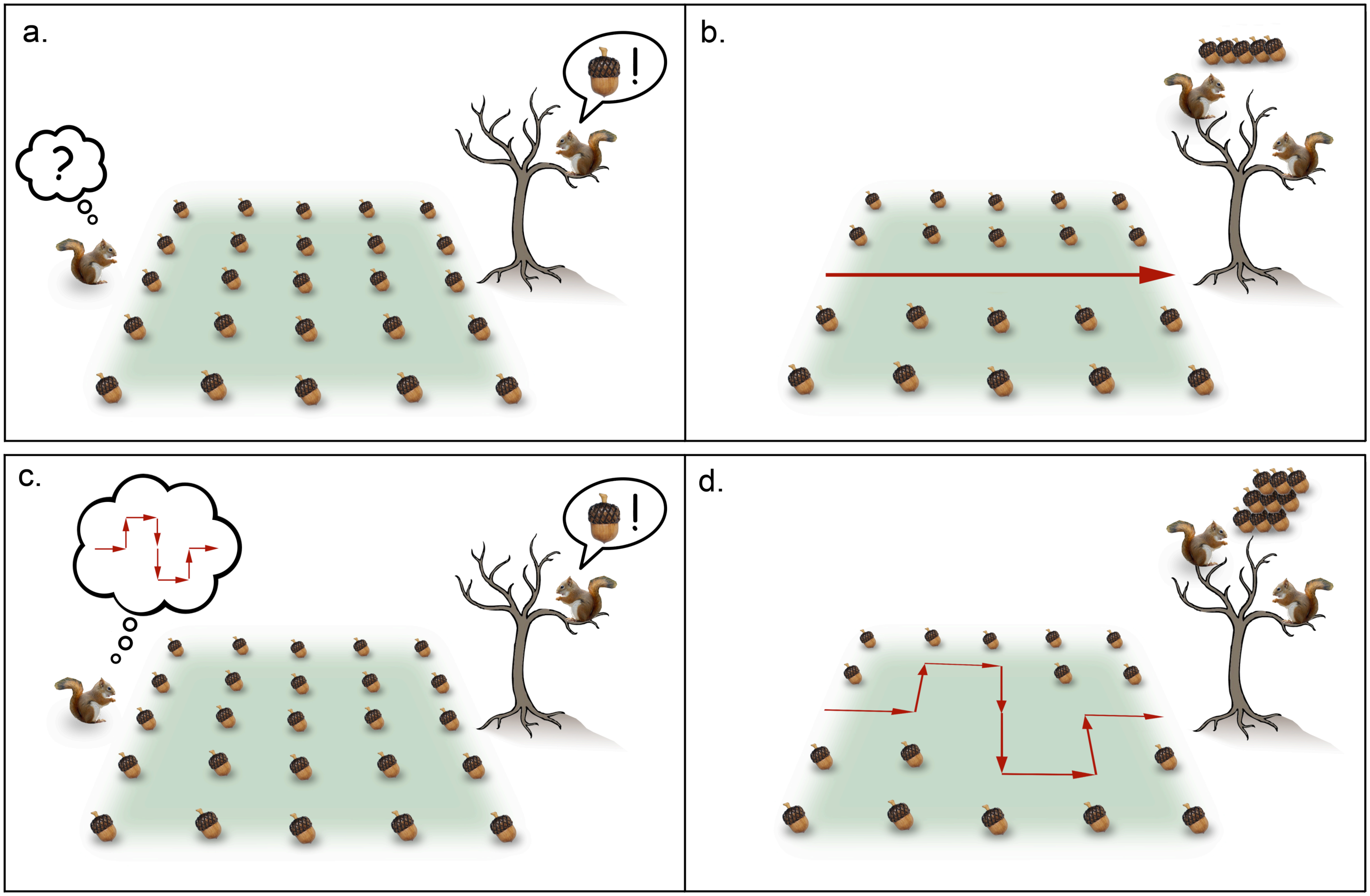



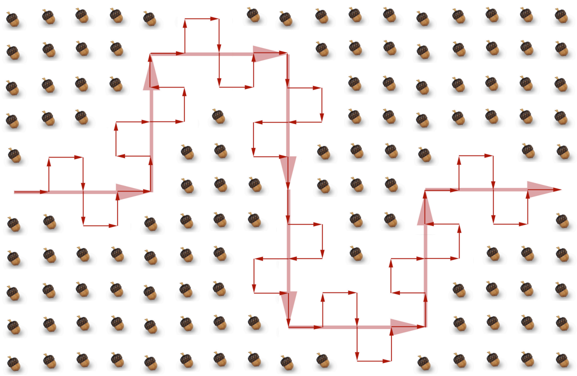
Convoluted paths are also created when animals forage, or seek out things in the environment. Let's do a visual thought experiment: imagine that you are a squirrel, scurrying around on the ground under a tree. Your mate is up in the tree, and she asks you to collect as many acorns as you can, and bring them up to her…in a hurry.

You think to yourself, "well, I could go straight to the tree and collect all the acorns in my path. That would get me to my mate quickly, but I may not collect many acorns." This is illustrated on the next page in panel (b). Just as you feared, when you arrive, your mate complains: "only five acorns?" You confess that there were only five acorns in your immediate path, and you grabbed them all up. The next day your mate asks you again to come back home to the tree and to bring back as many acorns as you can. This time you decide to take a few diversions on your way to the tree (one diversion to the left, and one diversion to the right). These diversions yield four more acorns, as illustrated in panel (d).

If you feel that you have a little more time on your hands (or rather, your paws), you might decide to make smaller diversions inside of your diversions – to grab up even more acorns. This is illustrated at the bottom of the page. These smaller diversions have the same shape as the bigger ones, but they are on a smaller scale.

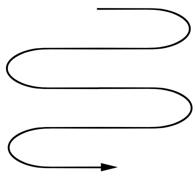
The result of this visual thought experiment is a two-level fractal curve – a curve that may look familiar to some of you: it is the boundary of the *quadric Koch Island* [15]. The visual diagram that the squirrel conjured up in panel (c.) is the "generator" for this fractal curve. The concept of *generators* will be explained in the next chapter.





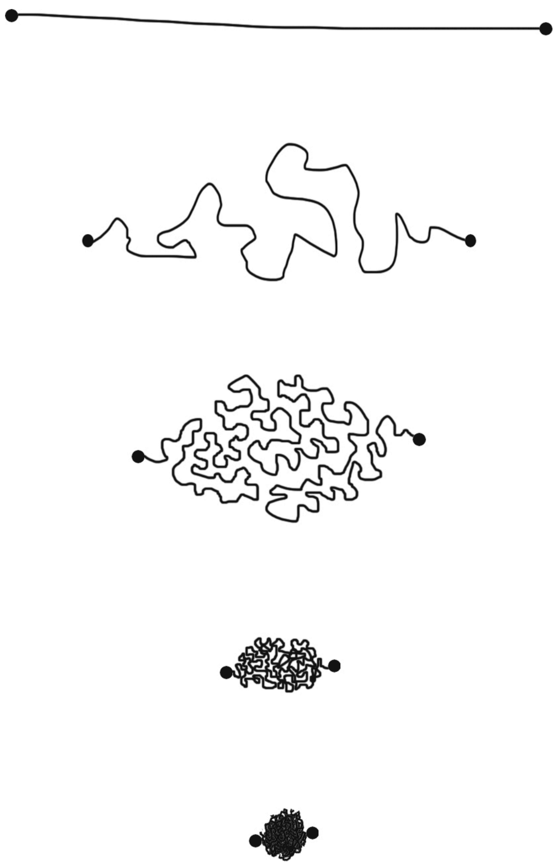
The diagram at left shows how the paths that make up the squirrel's left-right diversions each have their own miniature left-right diversions.

Now, you may ask: why doesn't our make-believe squirrel just scan the whole field of acorns one row at a time? Well, that's a valid way to collect a field full of acorns, but it's boring! (Our squirrel has a fractal mind.)



How many ways can you draw a curly line?

Infinite Strings



Imagine taking a piece of string, and coiling it up so that it takes up a small space. The picture at the right shows a string with black dots drawn at the endpoints. As we move these dots closer to each other, we allow the string to curl up.

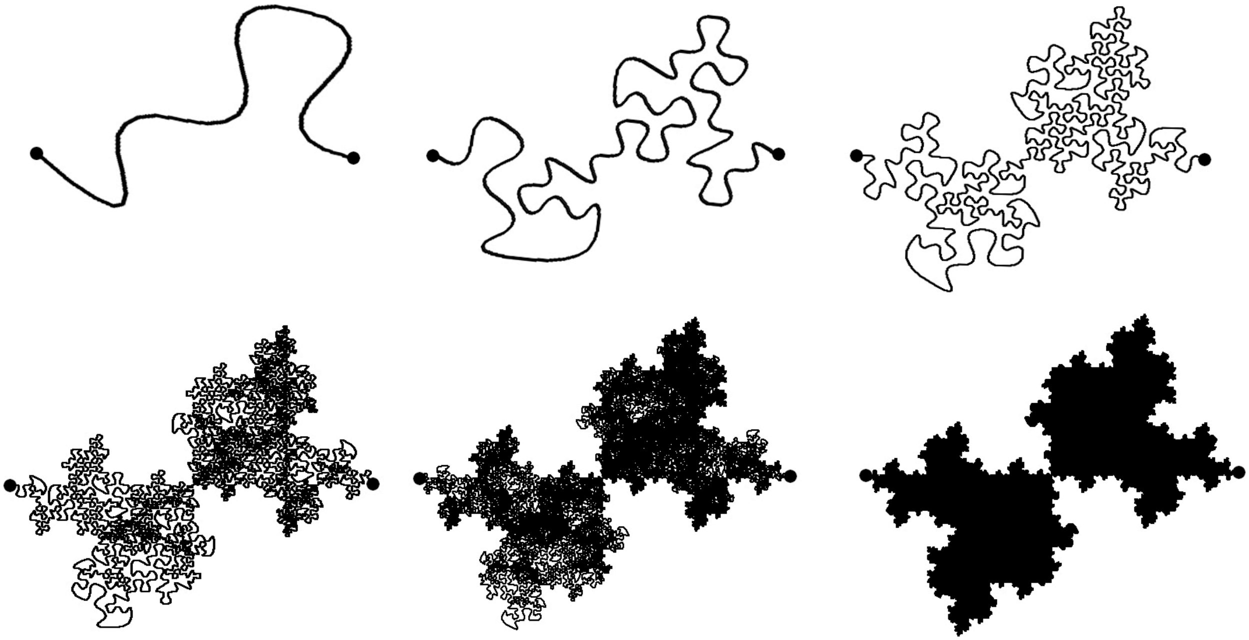
Imagine that our task is to make sure the string never touches itself or crosses over itself, even as we push the endpoints closer and closer to each other. Eventually of course, the string will get packed so tightly that it will have to touch itself. But that's only because physical strings have thickness.

This book is not about mortal pieces of string that reside in our physical world. It is about Platonic pieces of string. A perfect geometrical curve is infinitely thin, and so it can keep coiling up as the ends get closer together – essentially forever…and it will *never* touch itself.

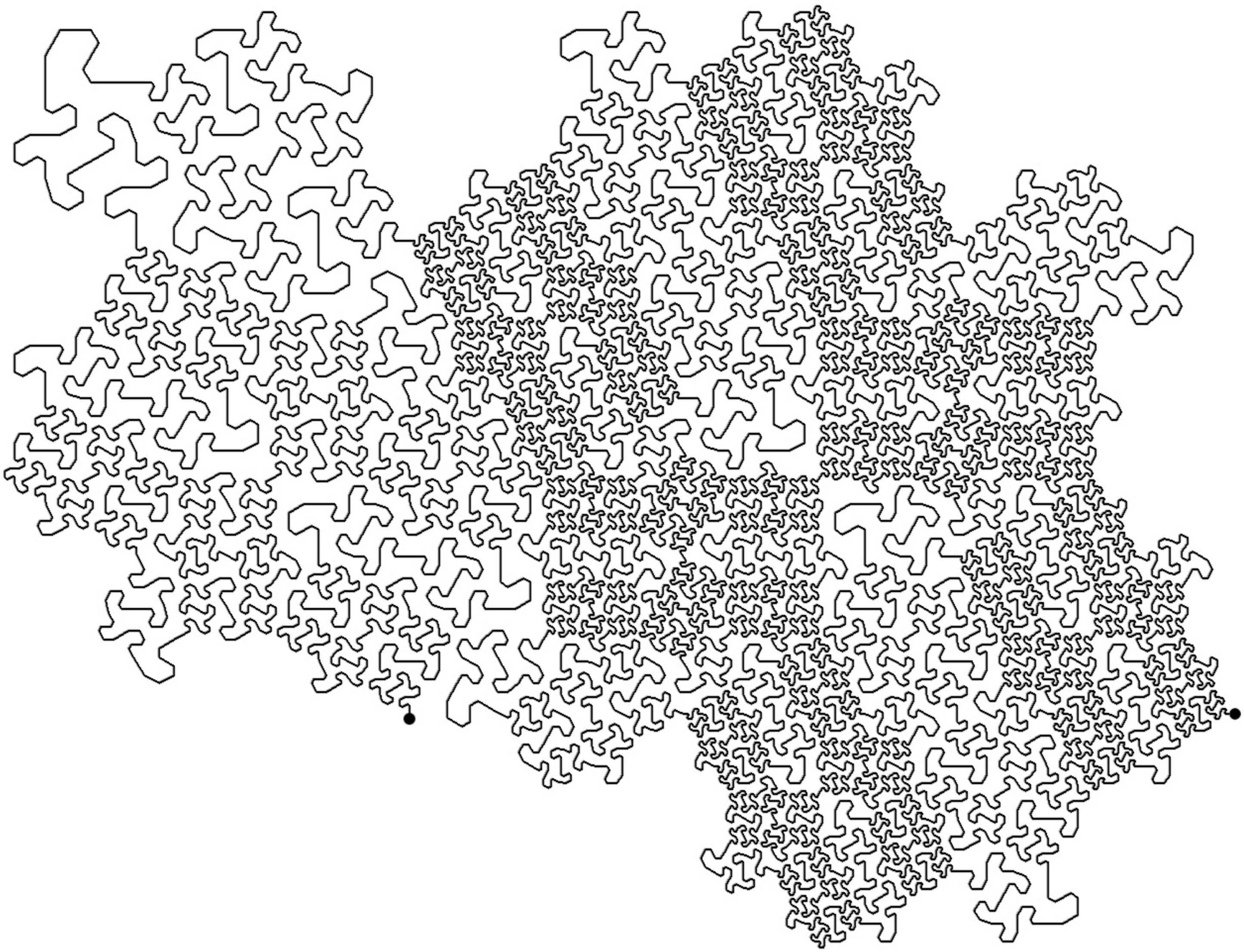
But that's no fun. After all, the end result of this process would be a blob of string so small that it would basically be a point. And it is well established in science that infinitely small things like points are extremely difficult to see.

So instead let's imagine that we keep the endpoints the same distance from each other, and that the string gets longer and longer, adding more and more bays and peninsulas to fill up space. This is illustrated below. I use the term "fractalize" to describe this process. It turns out that there are many ways that a curve can be fractalized as it gets longer. And we can capture some of these ways in simple, elegant geometrical processes.

The art of discovering these processes, and the categorization of these curves, is the subject of this book.

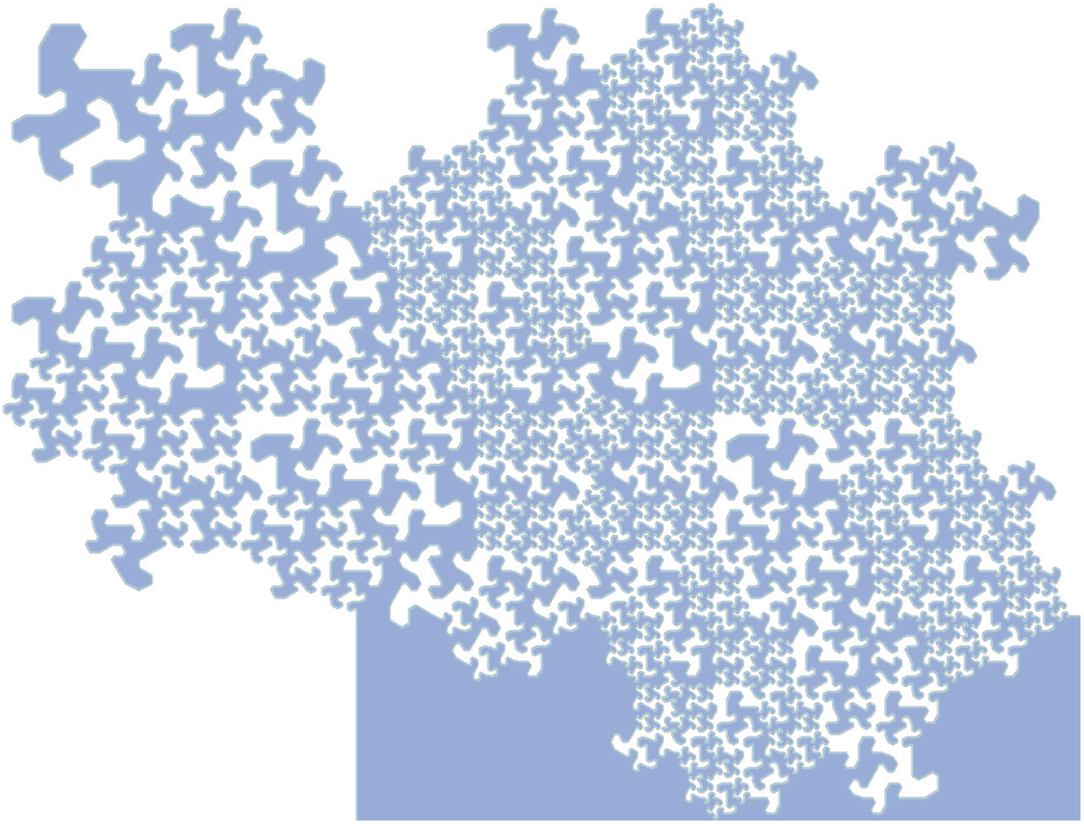


The drawing below shows one of the many fractal curves I have discovered. The two ends of the line are shown with black dots, visible at the lower left and lower right. Because of the particular scheme this fractal curve uses to fill space, there are unique self-similar patterns distributed throughout. The explanation for this phenomenon is explained later, as well as the genetic code used to generate this particular specimen.



Filling-in the Gaps

A fractal curve that doesn't cross itself can be described as the boundary between two highly-intertwined domains. Like sea and land, or like lungs and air, these boundaries can be highly convoluted. Let's take the fractal curve from the previous page, and fill it up with light-blue water, to see what happens…

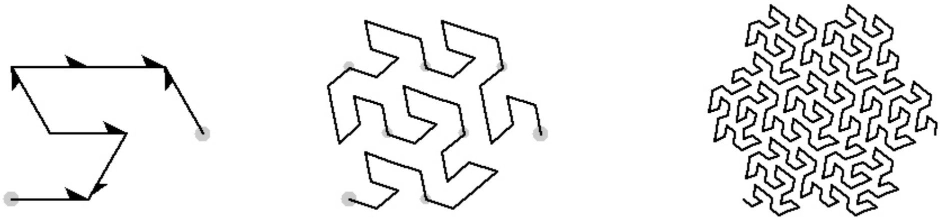


The curly lines have receded to the background and what we see instead are two domains: one white, and one light blue. If this were a map of canals in a seaside village, you would use this to determine a route to paddle your gondola from one part of the village to another. As a general rule, I prefer to let the beauty of the curve itself shine through, but in many cases, the shapes between the curves are just as interesting. You will see that I use this colorizing technique for many of the fractal images in this book.

Filling Your Brain

The evolution of brain folds is roughly correlated with advances in animal intelligence. Here are three brains for your viewing pleasure: from left to right: rat, monkey, and human (these images have been scaled relative to each other).

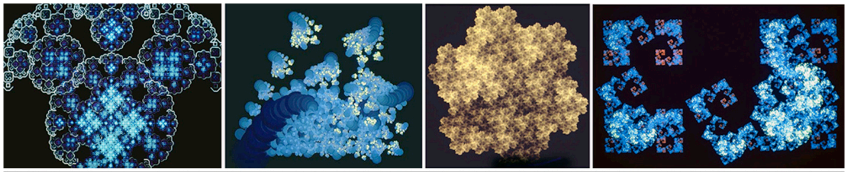
I would like this book to cause a few more folds to grow in your brain. And I would like that growth to be joyful. Curves that completely fill up a region of a plane are called “plane-filling curves”. They are familiar to fractal-lovers: mathematicians, geometers, and artists of the Escher ilk. For many of us, the reason we love these curves is not because they answer questions like how to pack leftover spaghetti into a jar, or how to build a maze for your pet rat. The reason is because we find them beautiful – visually and intellectually.

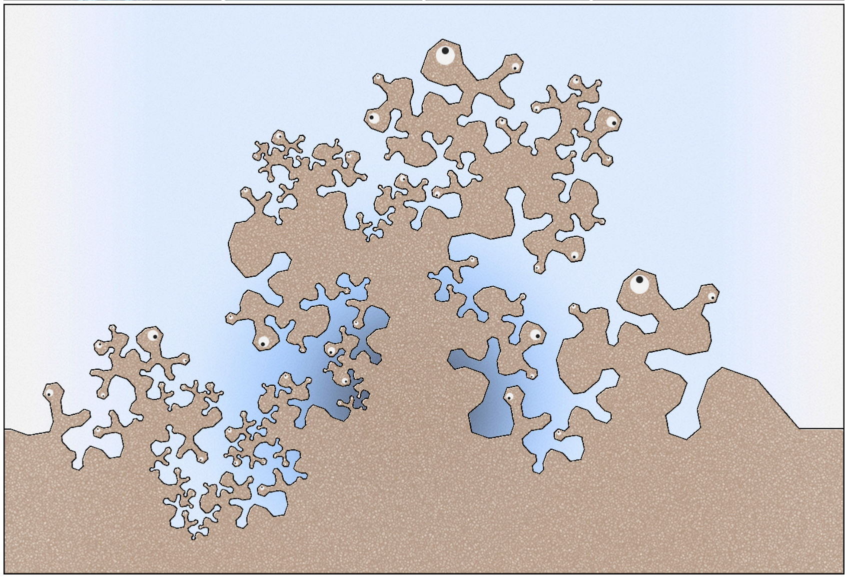


Although these kinds of curves have been created in the past, they were generally considered as mathematical curiosities, or worse, “monsters” until, in the 1970’s, Benoit Mandelbrot coined the term “fractal” and brought together many of these beasts under one umbrella; as a unified mathematical discipline. In his book, The Fractal Geometry of Nature [14], Mandelbrot referred to plane-filling curves as “Peano Curves”, in reference to Guiseppe Peano, an Italian mathematician who described such curves in the 1890’s [18]. Mandelbrot’s book is the foundation for many of the concepts and terms I use in this book. But I shall extend these ideas and terms, and put them in a new context – a context in which plane-filling fractal curves can be explored without end – and with beautiful results.

A Definition

Now is a good time to give a more formal, more general definition: a "space-filling curve" can be described as a continuous mapping from a lower-dimensional space into a higher-dimensional space. In traditional mathematics, a *curve* is described as a topological space that is *homeomorphic* to a line: if you magnify a small region of the curve, it looks like a straight line. The higher the magnification, the more it looks like a straight line. But…fractals came along and changed all that! Consider a curvy line that has an infinite number of curls, bends and folds…at every level of magnification. And consider a curve that has such a serious case of *horror vacui* that it visits *every point* in a planar area in its path from start to finish. This is what we mean by a "plane-filling curve". It is the topological equivalent to a planar shape covering that same region. So you can think of this curve as being a particular way to describe that planar shape. In other words, it provides a mapping between a line and a 2D shape. Mandelbrot used the term "sweep" to distinguish a plane-filling curve from any old 2D planar shape, implying the progression of a path that passes through every point in that shape over some period of time. (A more thorough overview of space-filling curves is given by Hans Sagan [20]).



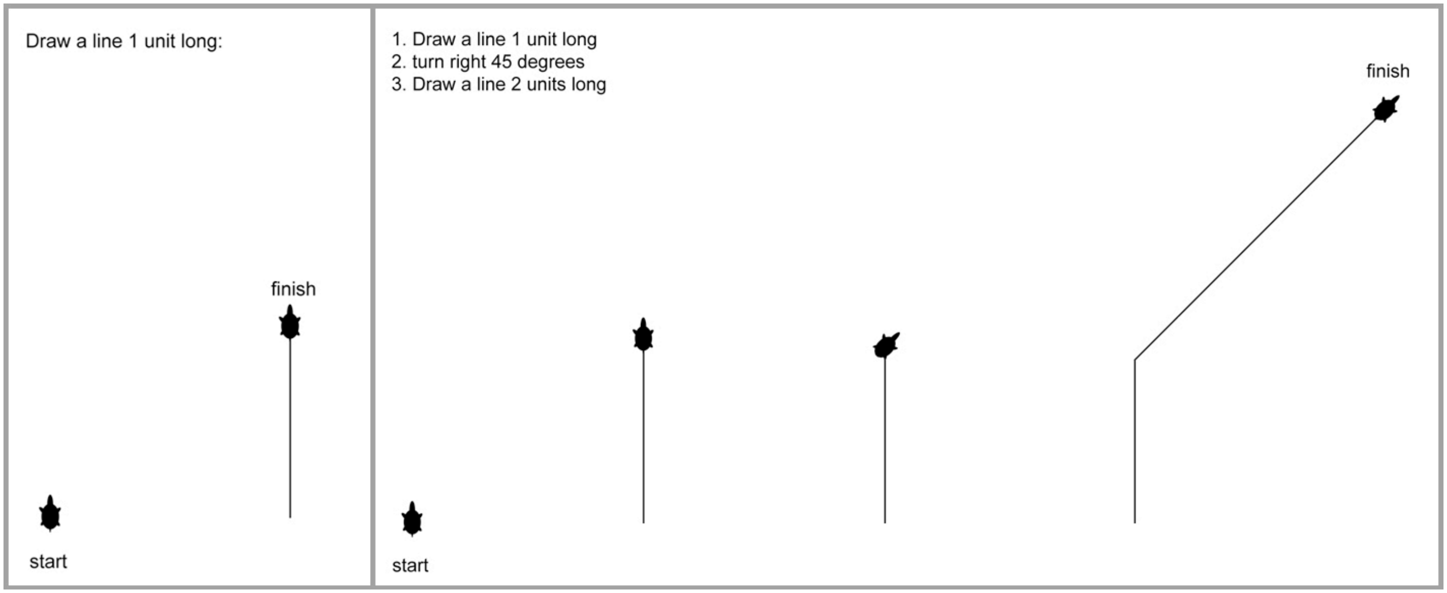


I have been exploring fractal curves now for about twenty-five years, and I have drawn several hundred pictures (probably more than a thousand), in my lifelong search for fractal curves. I have also developed genetic algorithms and other computational search techniques to find new specimens. This book shows the culmination of my search for plane-filling curves. I have accumulated over 200 specimens and organized them according to a taxonomy of fractal curve families. Each specimen is shown with a unique genetic code. And I have included several color images of what I consider to be the most striking specimens. Many of these – I have reason to believe – have never been seen before, and I am delighted to introduce them to you.

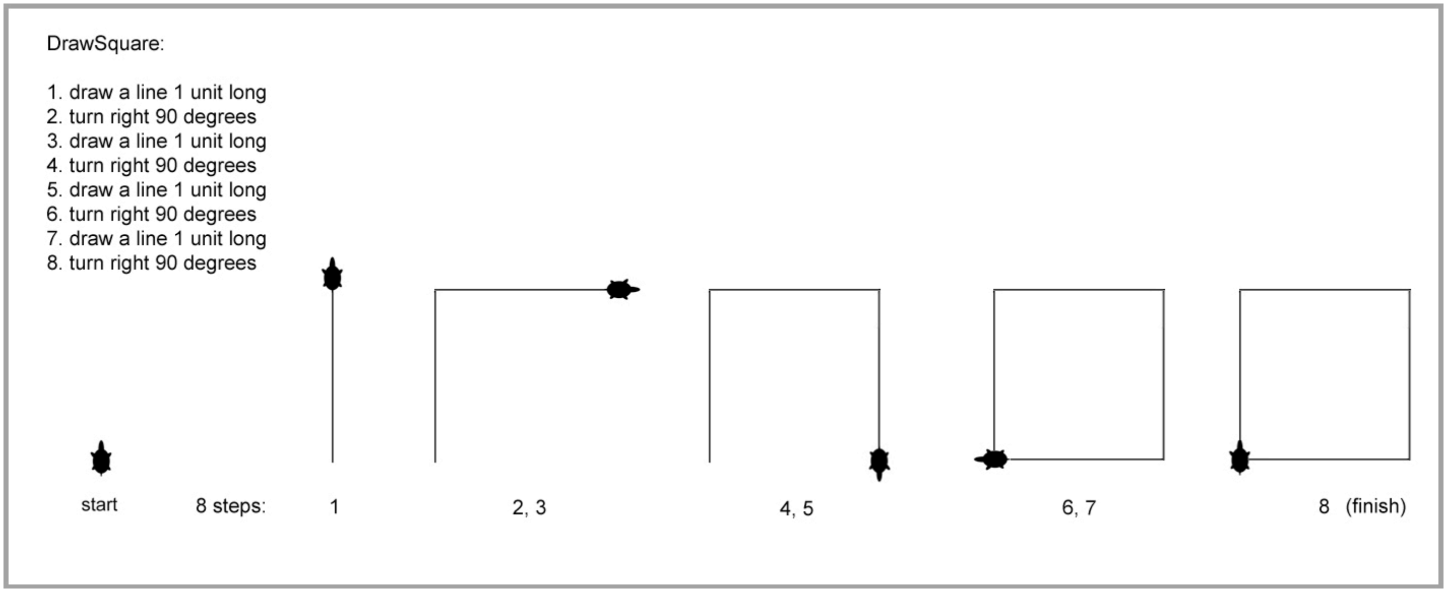
2

A Very Patient Turtle Who Draws Lines

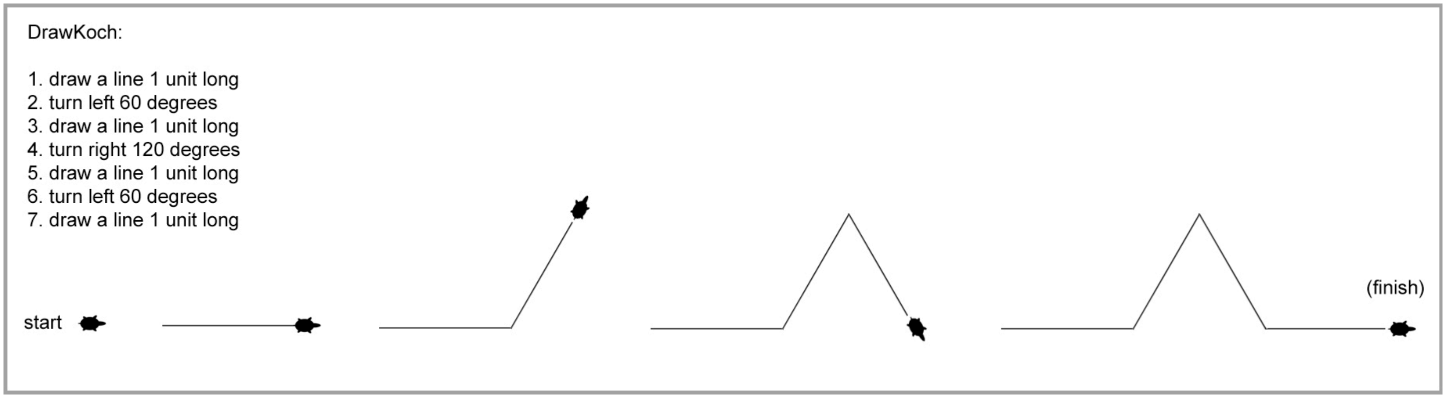
Before we go on, I want to say a few things about process. And for that I will introduce you to a helper who will be demonstrating the principles used for drawing fractal curves: the *LOGO turtle*: a key feature of the LOGO programming language, used in education [0]. The turtle has quite a distinguished history. It is used to help young people learn about programming, math, and graphics. I am of a generation of people who learned about computer graphics and programming with the help of the turtle. I am excited to show you how the turtle can be a bridge that connects simple, visual ideas to advanced mathematical concepts, including the very complex and beautiful world of fractals. Let's follow the turtle through a few demonstrations. It is shown as a little black image of a turtle, as seen from above. It can move forward, rotate its body, and draw lines. Here are some basic actions that the turtle can do:



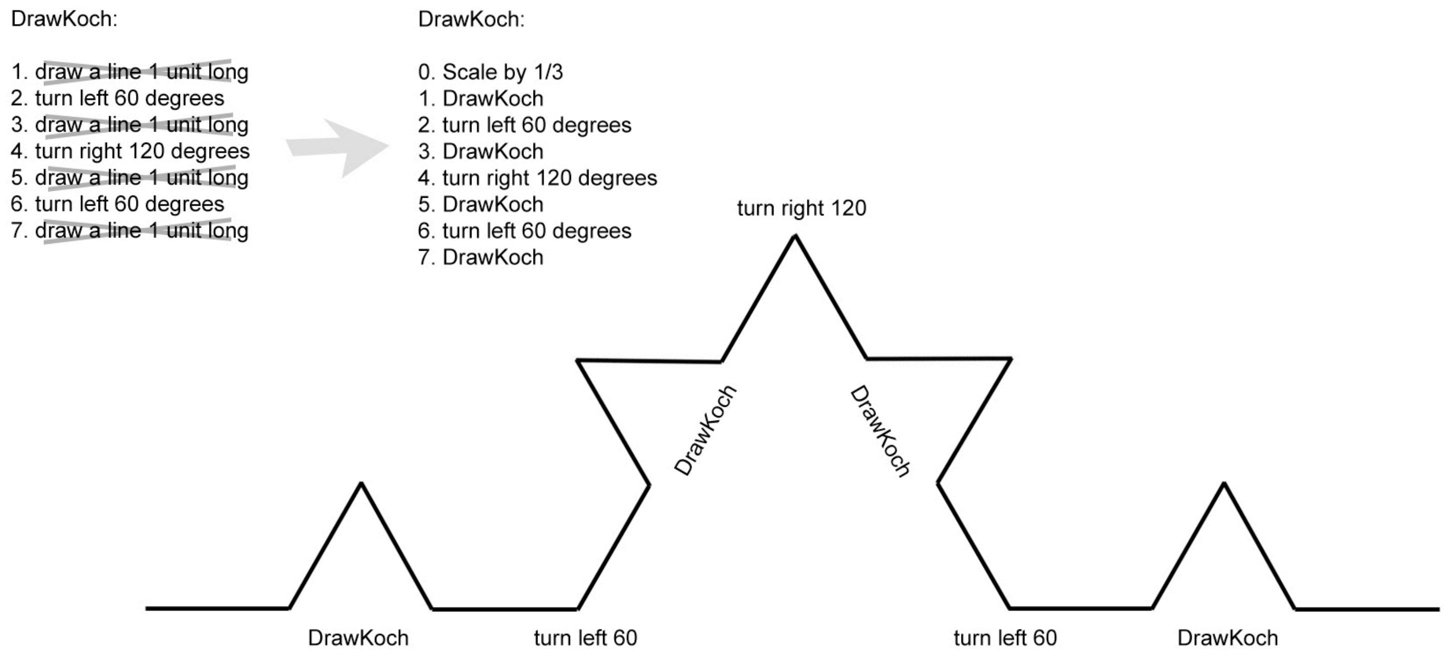
The turtle can perform a set of commands as one action – as a *procedure*, such as “DrawSquare”.



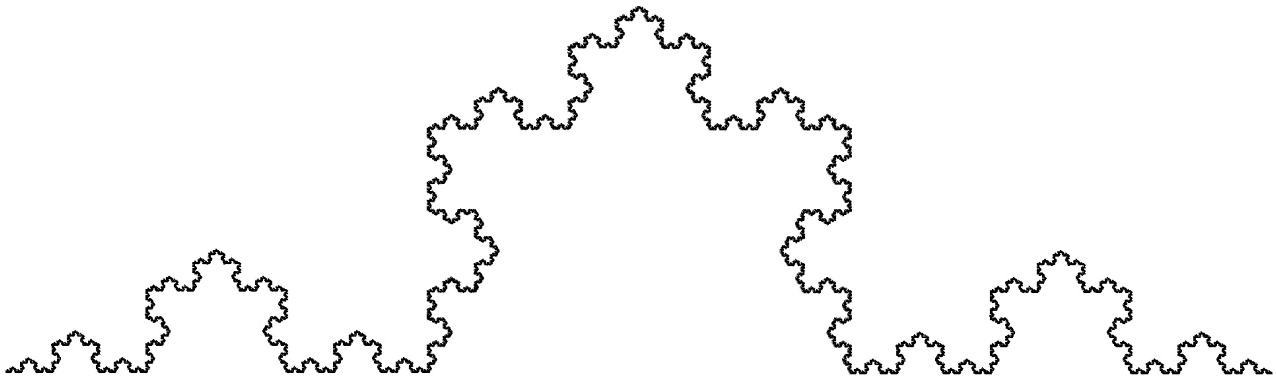
Now, the turtle will demonstrate a shape that is the basis of a well-known fractal curve: the *Koch curve*. First, the turtle will rotate 90 degrees, so that it is aiming rightward. Then it will perform the procedure “DrawKoch”:



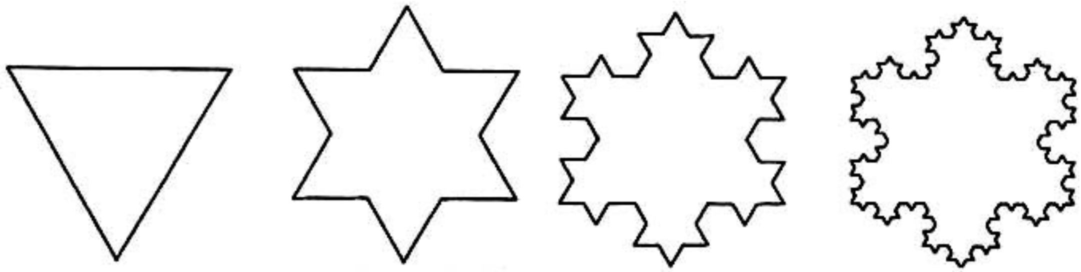
This procedure, *DrawKoch*, can be modified so that it is scaled to one-third its original size, and the steps that draw lines (steps 1, 3, 5, and 7 above) are replaced with copies of the whole procedure, as shown on the next page.

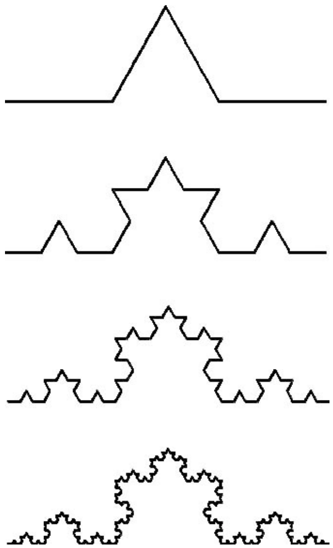


This means that instead of the figure consisting of four line segments, it consists of four smaller copies of itself. Now, if you think about this for a moment, you will realize that there is a paradox: each of these copies will then have to consist of copies of themselves as well! And so on, and on. What we have is an infinite regression of copies of copies of copies…. Presto! We have a fractal. The problem is that it takes an infinite amount of time to draw an infinite number of lines, so we choose a cut-off point where recursion stops. (The picture below stops at 5 levels.)



By the way, this curve is usually referred to as the “Koch Snowflake Curve”, from the original description in 1904 by the Swedish mathematician Helge von Koch [12][13]. He showed the curve as a progression of applying triangular bumps upon bumps onto the perimeter of a triangle. It looks a bit like a snowflake.





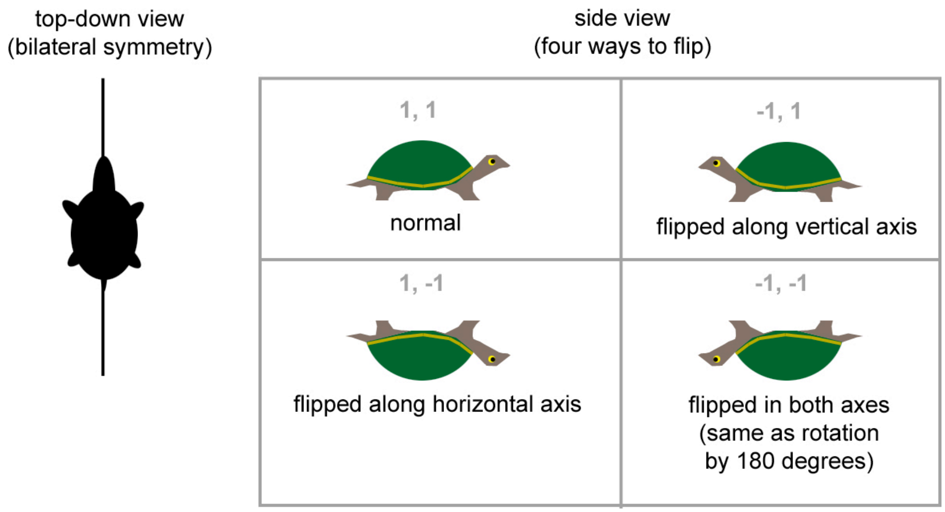
The Koch snowflake is the equivalent of three Koch curves connected together as if they were the sides of an equilateral triangle. Mandelbrot referred to this triangle as the “initiator”, and the curve that is placed onto the three sides as the “generator”. In this book, I am mostly concerned with generators: that is, what happens in-between the two endpoints of a single fractal curve.

I will not get into the nuts and bolts of programming fractal curves here. But I do have to explain some aspects of the fractal algorithm: When I make a fractal image, I can control the number of levels of which the copies are made. In other words, I can draw the Koch curve at level 1, in which case it just consists of four line segments, or I can draw it at level 2, in which case those lines consist of smaller copies of the Koch curve (scaled to one-third of the length of the segments of level 1), making the whole curve a figure of 16 line segments. Level 3 would consist of 64 line segments, and level 4 would consist of 256 line segments, and so on. To be consistent with Mandelbrot’s terminology, let’s call this fractal curve a “teragon” (as in the “5th teragon” or “12th teragon, and so on). Also, throughout this book I will be using the term “fractalization” to denote the process of increasing the fractal level from one teragon to the next – the process of replacing each segment with a small copy of the generator.

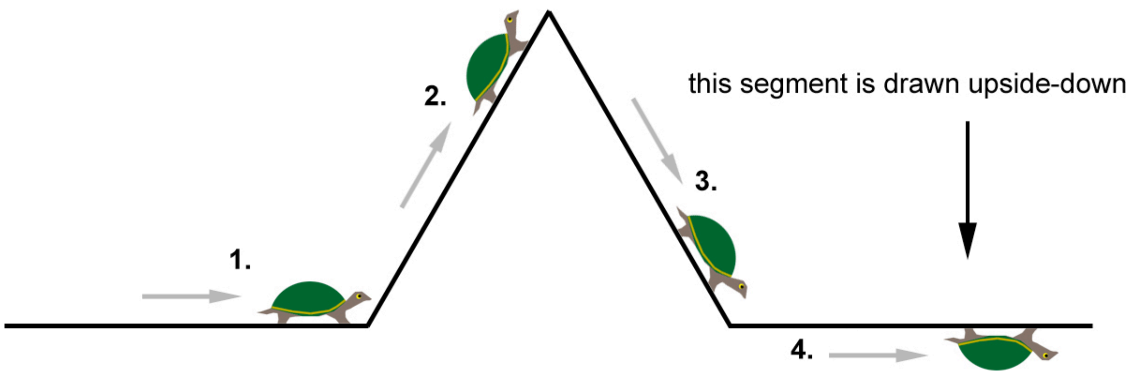
Let’s assume that the algorithm that draws a fractal curve automatically scales the smallest copies so that the distance from endpoint to endpoint is always the same, no matter how many levels are used, and that the lengths of the line segments always scale appropriately, just like the example shown on this page at right.

The Turtle Flips Out

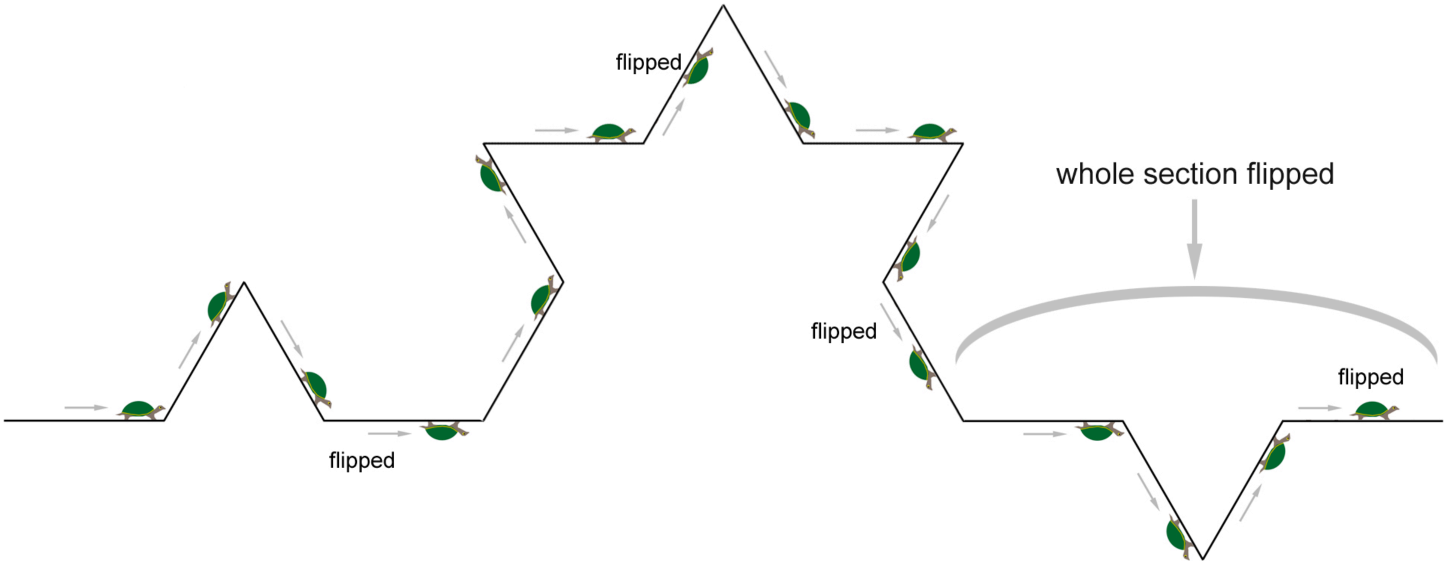
Now, there is an important aspect of the fractals that our little turtle will be drawing; and that is the fact that the line segments in a fractal generator can be "flipped" in various ways, to make the copies do clever tricks (which are important for enabling the many varieties of plane-filling curves that will be shown later). But before I show you these tricks, I need to create a new view of the turtle, to make it easer to explain. This view comes in four flavors, as shown below:



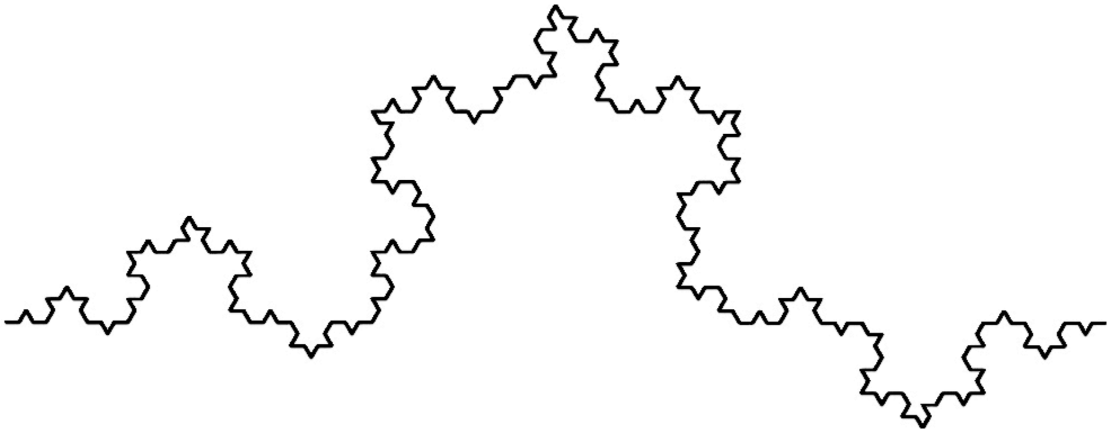
Now that we have these different views of the turtle, consider an image of the turtle drawing the Koch generator like this:



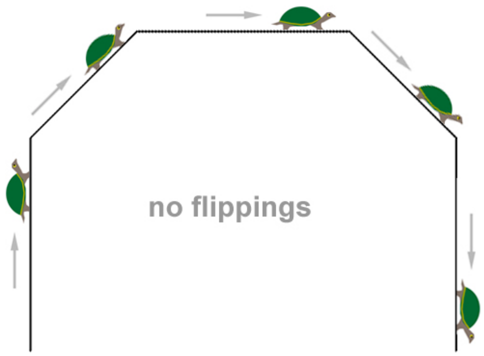
The implication of this flipped segment is that the copy of the Koch generator that is drawn in place of that segment is upside-down.

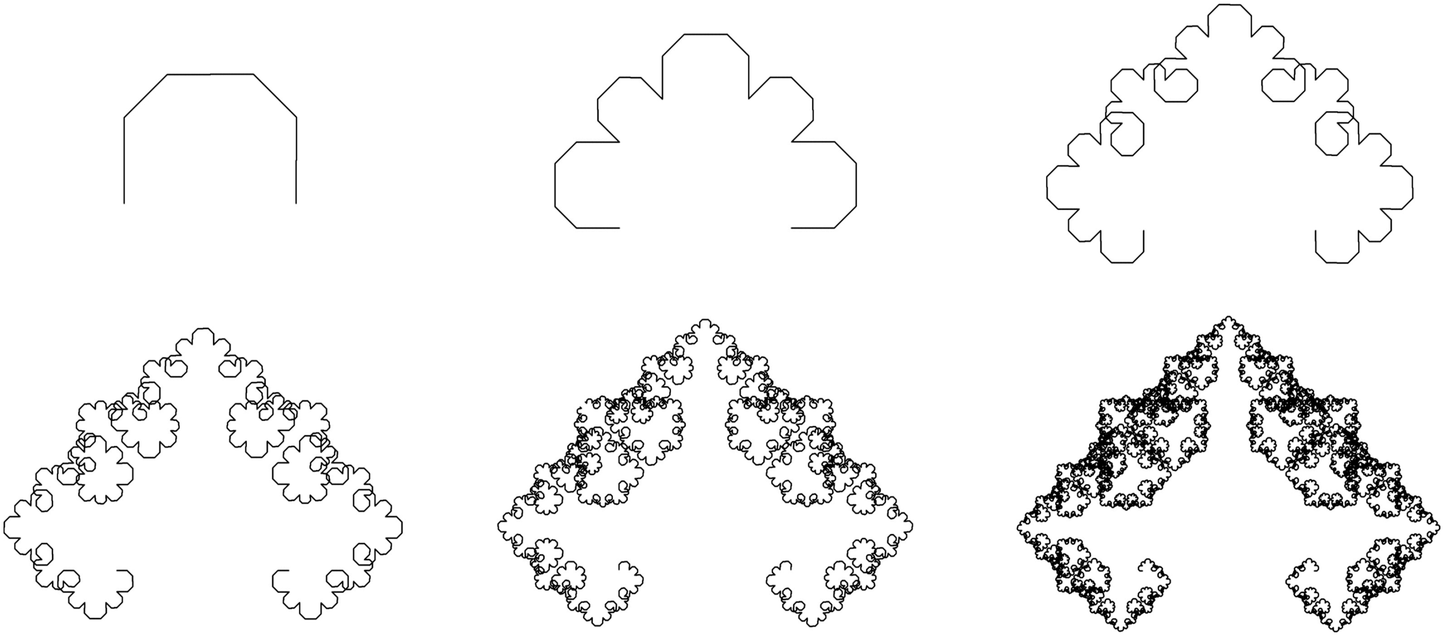


Now let's fractalize this to teragon 4. Well, this is no longer the Koch curve we know and love! It has changed quite a bit, due to this single flipped segment in the generator. This flipping gets propagated to all the sub-copies, and it makes its effect all they way to the small details of the fractal (although not in the smallest level).

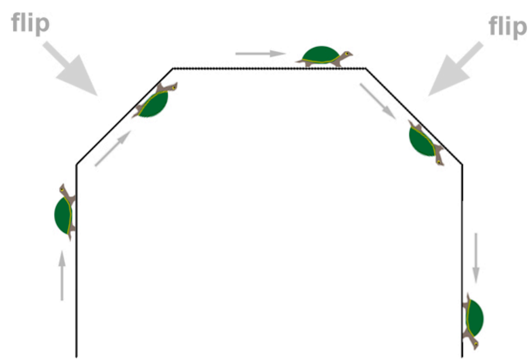


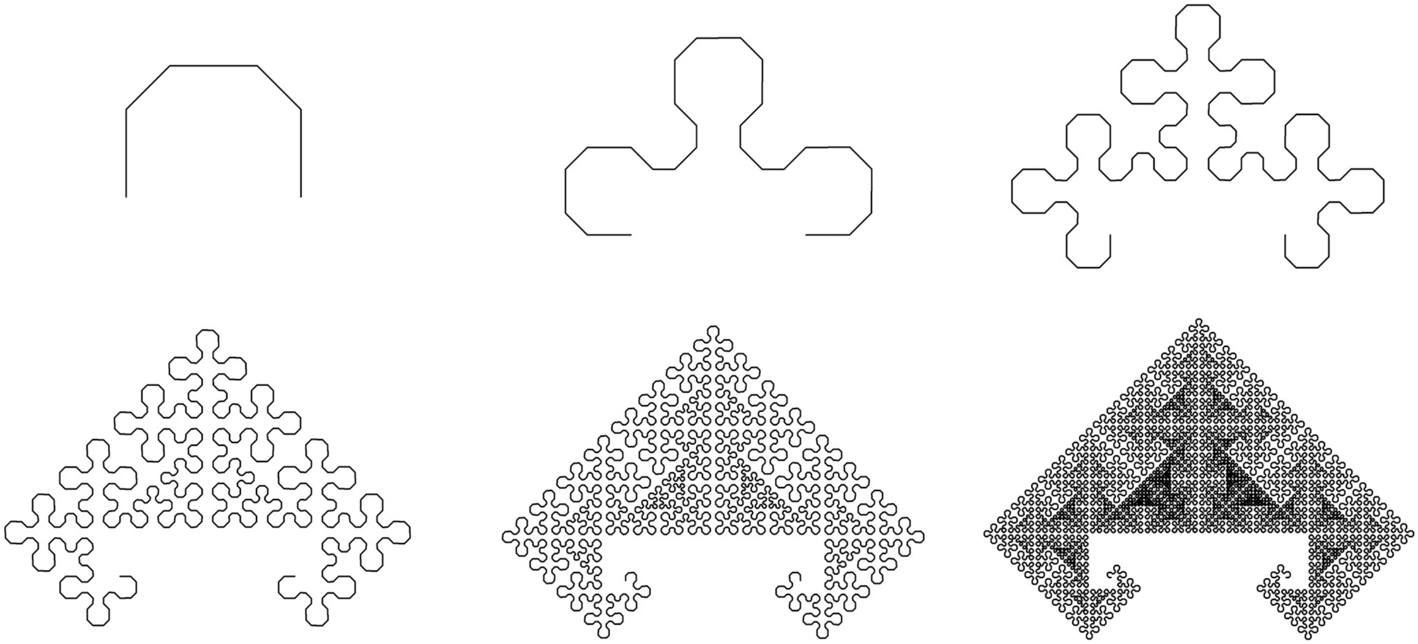
How can this flip trick be used to make a plane-filling fractal curve? Glad you asked. I will show you. But first, consider the following fractal generator with *no flippings*, and its first 6 teragons:





That is an interesting fractal indeed. But it crosses itself all over the place – starting at teragon 3 and increasingly for each level. It turns out that the turtle can do some flipping on alternating segments (the two diagonal ones), to transform this into a plane-filling, self-avoiding, fractal curve. Here it is with the first 6 teragons:

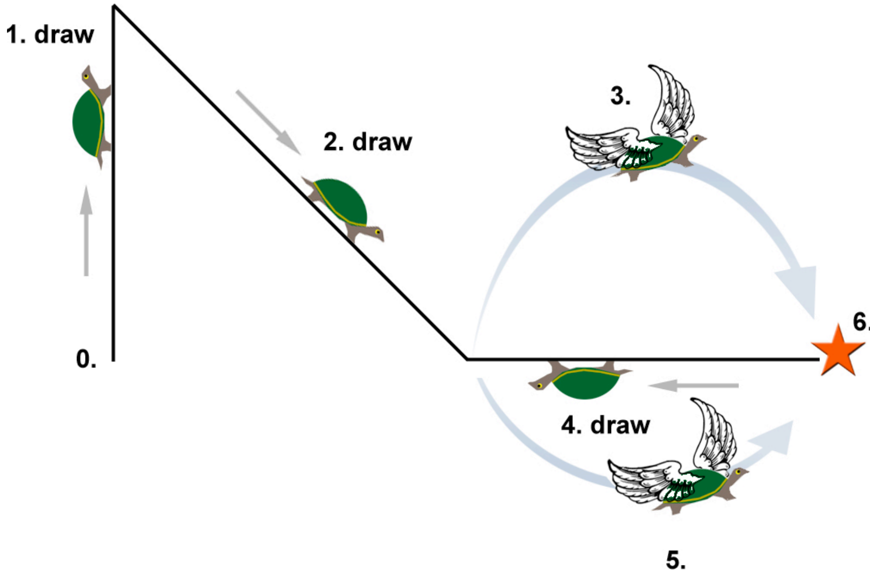


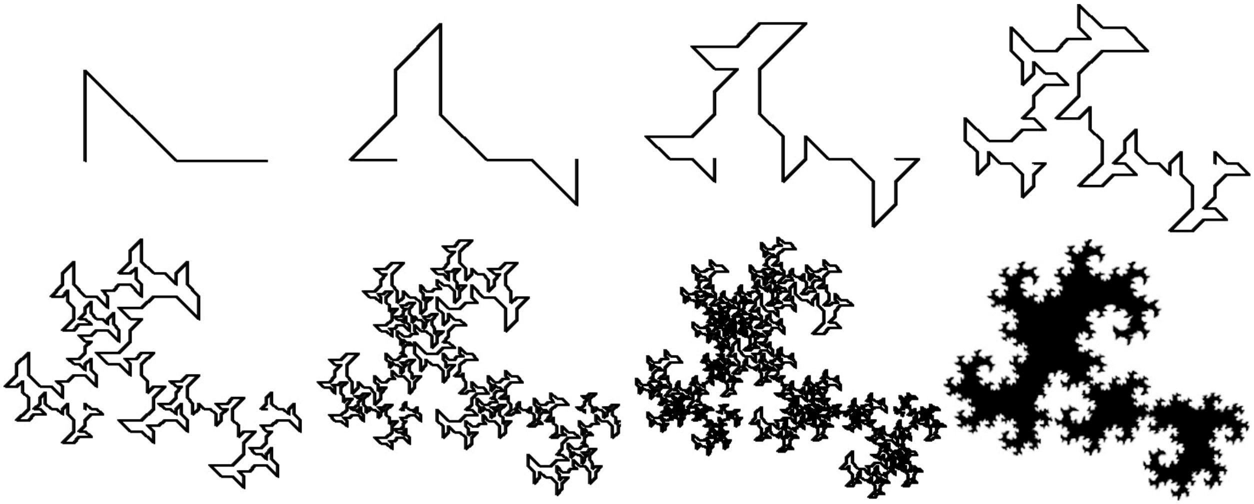


With these two flippings, this curve can be fractalized forever and it will *never* cross itself.

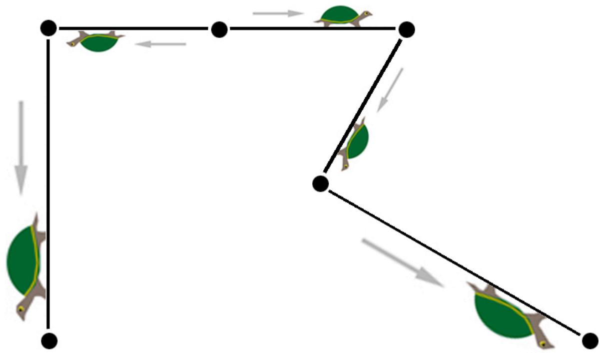
A flipping along the vertical axis is a bit more tricky to explain, because it implies that the turtle has jumped to the end of the line and drawn it backwards. This concept might be easier to grasp for those of you who have used the LOGO turtle, because the turtle can be told to *move* to a position (without drawing a line).

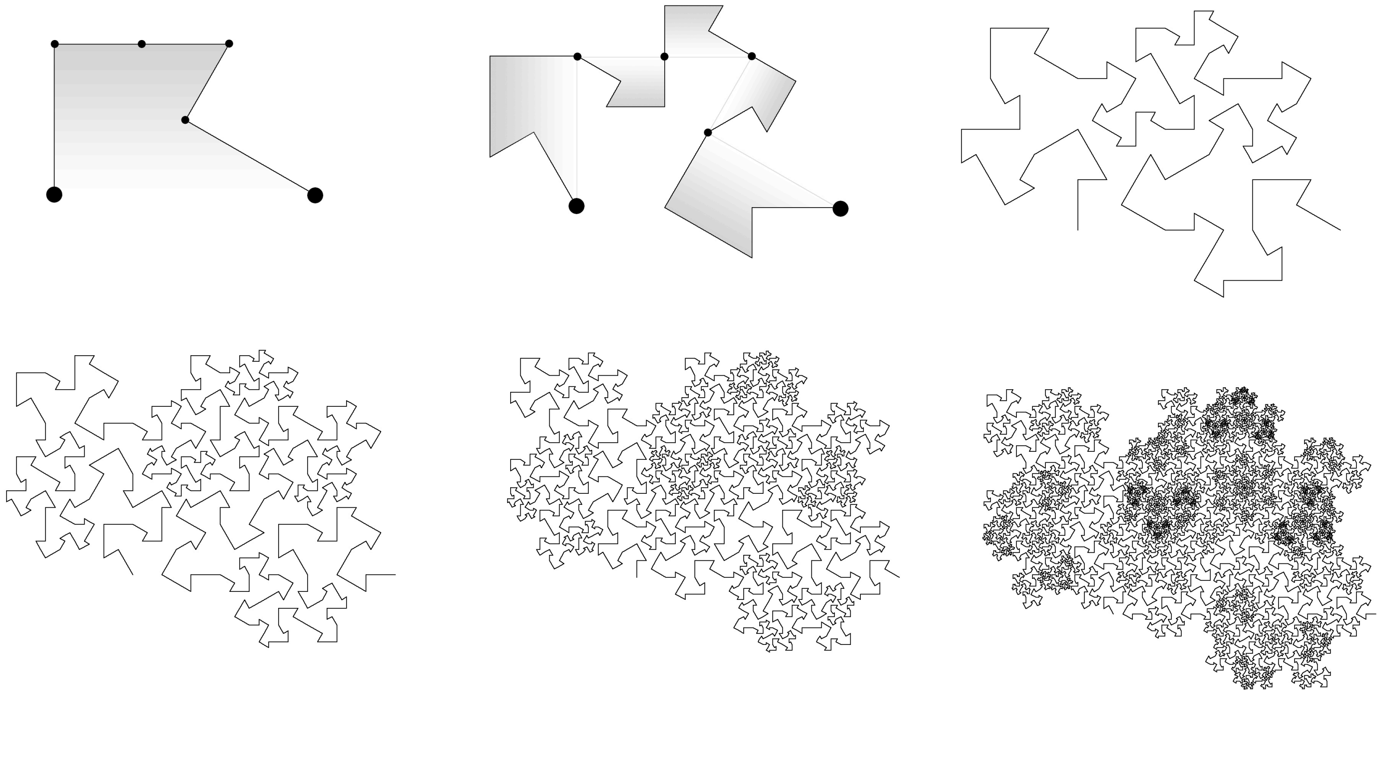
To illustrate this in a playful way, imagine the turtle drawing a 3-segment fractal generator in 6 steps, illustrated at right. In order to draw the third segment, the turtle flies to the end (step 3), draws the line backwards and upside-down (step 4), and then flies back to the endpoint again (step 5), ending at the correct finishing location (step 6). What is the fractalized result? It is a cool self-avoiding curve, called the "Dragon of Eve". Notice the strategically-oriented bump at the bottom-right of the second teragon – this is critical to making The Dragon of Eve a self-avoiding curve.



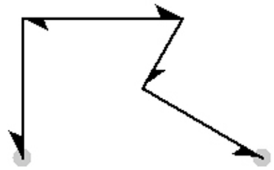


Now let's look at a plane-filling fractal curve that takes advantage of another kind of flipping. (In this example, I have shrunk the turtle in the top three segments, to indicate that they are half as long as the two segments at the bottom.) You may find it difficult to see exactly how these flippings allow the curve to fractalize the way it does. So to make it a little bit easier, I added some visuals to the first two teragons.



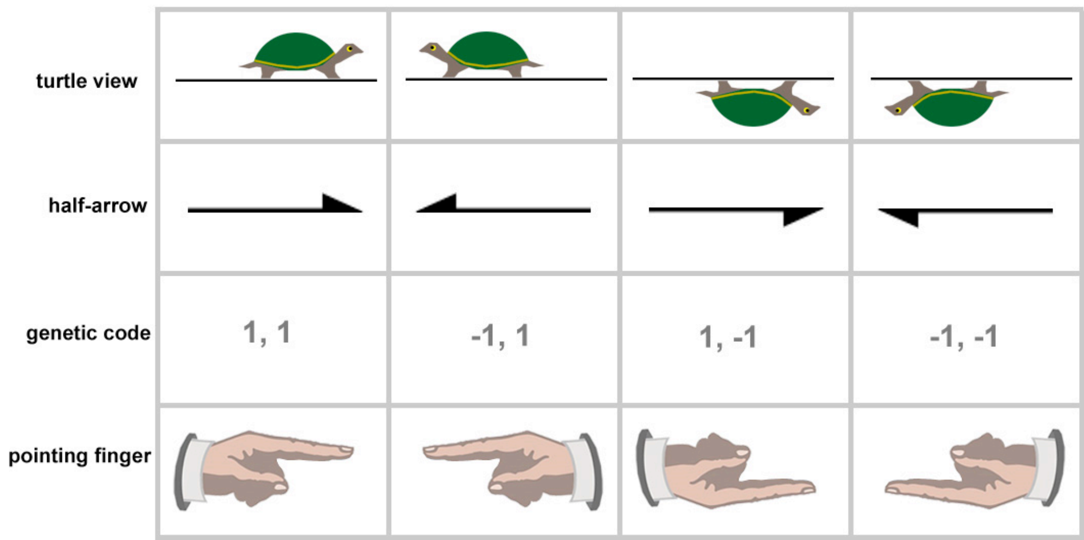


Pictures of flipped-out turtles are cute, but I will not be using them in the diagrams throughout this book. Instead I will be using half-arrows because they are more compact. So, the generator we just saw would be drawn like this…



In addition to representing these flippings visually with a half-arrows, they are also specified using a genetic code, as shown in the table below. At the bottom is a representation using a pointing finger. I will use this in a few examples.

Flip Representations

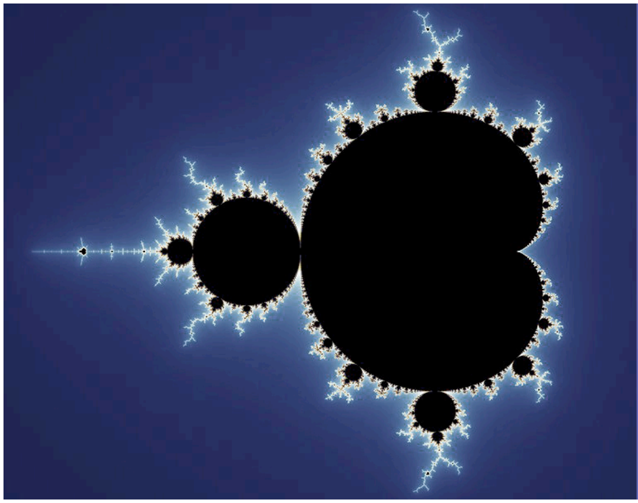


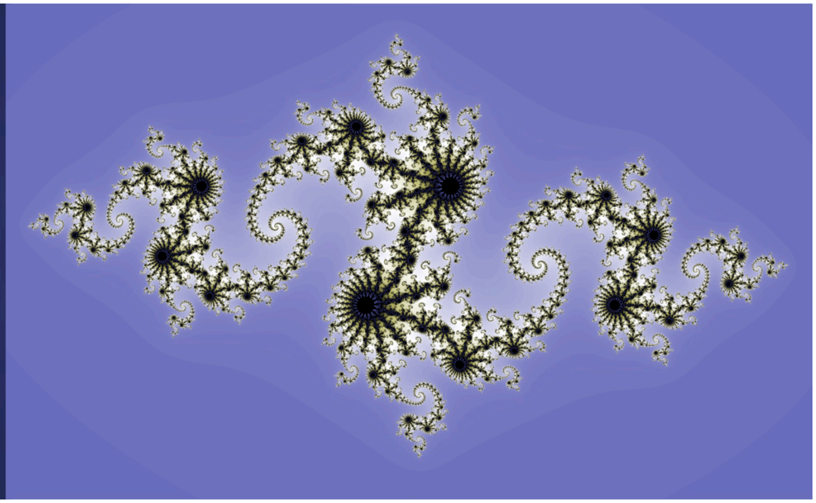
You may still be having difficulty imagining how a turtle drawing lines can create such shapes. You might prefer to understand the process in terms of transformations; copying shapes and then translating, scaling and rotating these copies in various ways to create new fractal levels. If so, you are in luck: in the next chapter I will show you some fractal generating techniques that are based on this idea. These techniques will help flesh-out in more general terms how fractals are generated, and it will put my particular scheme into a larger perspective. This will give some context for the more than 200 specimens archived in this book.

3

A Taxonomy of Fractology

There are many (*many*) kinds of fractals. The most popular fractals are the images generated with math equations iterated in the complex plane, such as the Mandelbrot Set and the Julia Sets, and all their amazing variations and magnifications.





These fractals are generated by calculating a mathematical function for every pixel location of a 2D grid, using the x and y coordinates of the grid as input values, to determine a color value. Mandelbrot and Julia sets use complex number equations. They have been studied extensively, and reproduced with endless variety on the internet.

Fractals of this type are beautiful, complex, and amazing. But they do not easily lend themselves to learning the basic geometry of fractals, as a form of visual construction. This is why I am so enamored with self-similar fractal curves – as tools to think with; as visual objects to apply various geometrical ideas and processes from other domains. The process

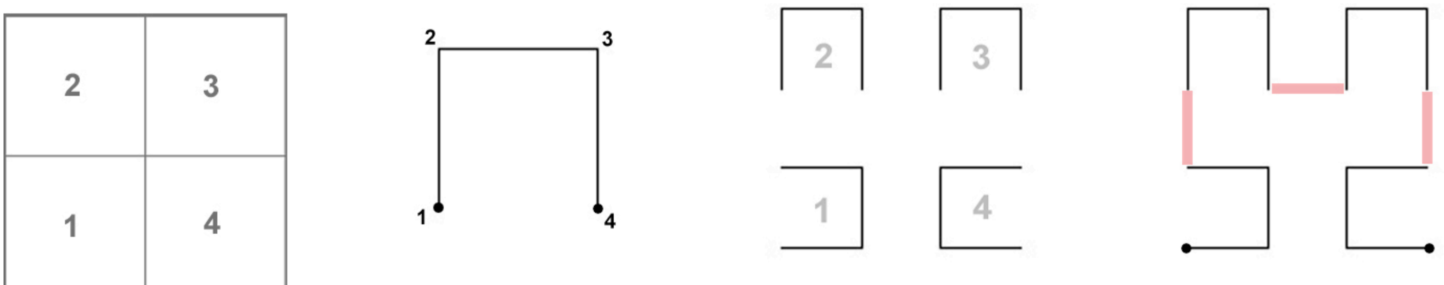
by which they are constructed can be explained to non-mathematicians. And if you are curious enough, and especially if you are willing to do a little bit of programming, it can lead to discovery upon discovery.

The fractal generation technique that I use in this book is a variation of “Koch construction”. It is based on the idea of taking a figure consisting of connected segments, and recursively replacing each segment with a copy of the whole figure. This is how the Koch curve is generated, and it is also how an infinite number of other possible fractal curves can be generated. Now, before I go on, I want to describe a few of the methods that have things in common with Koch construction. In some cases, these might be considered *variations*, or different approaches, to Koch construction. They are:

- Node-replacement curves
- Iterated function systems (IFS)
- Branching Fractal Trees
- L-Systems

Node-Replacement Curves

One of the most well-studied techniques for generating plane-filling curves is a process that is similar to Koch construction, except that it is based on *node replacement* instead of *edge replacement*. This technique has been explored by mathematicians and hobbyists for centuries. Peano, Hilbert, Moore, and Wunderlich, are among the mathematicians who have described such curves. Below is an example of using this technique to generate a well-known curve with a generator of four nodes: the Hilbert Curve [9]. It is based on the fact that a square can be tiled with four smaller squares. Take a look at how this simple generator is copied four times, transformed, and connected to form the shape at right:

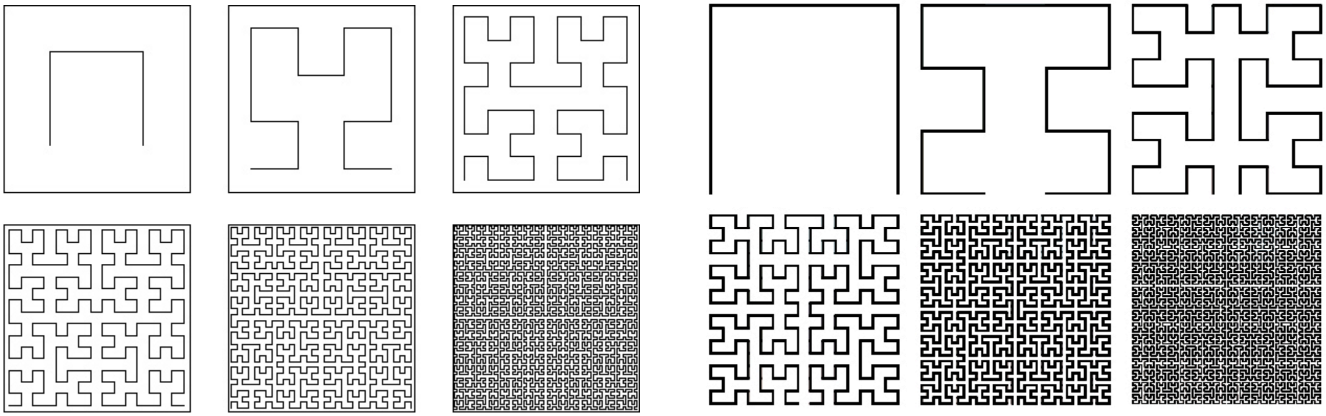


generator

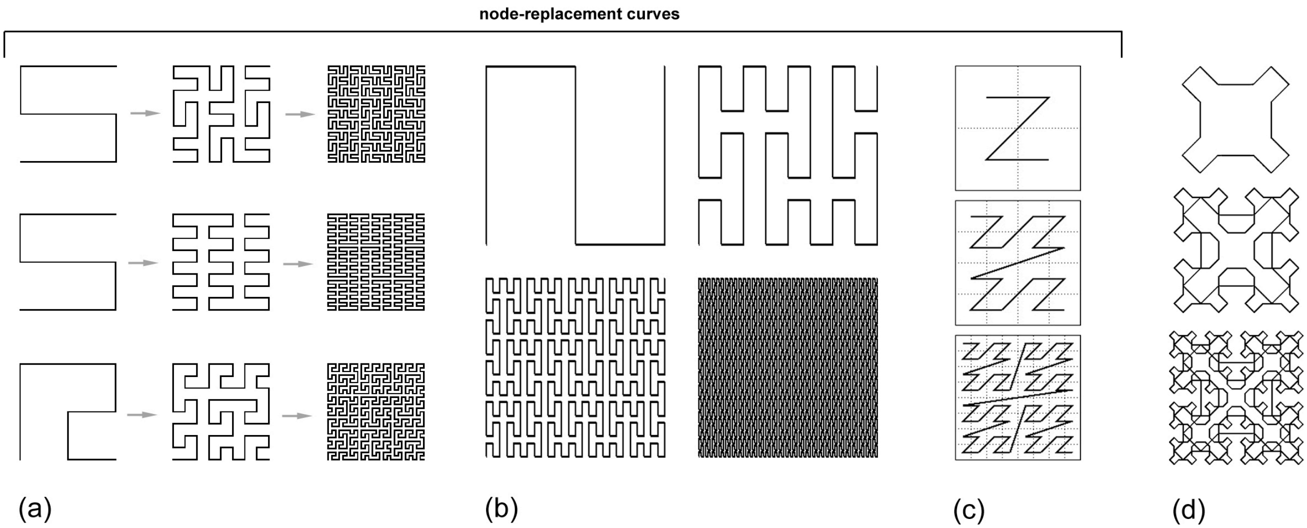
four copies, transformed

connected

Now let's see six progressive fractalizations (with node connections added). The Hilbert curve is shown at left. On the right is a variation called the *Moore* curve.



Node-replacement curves are generated by replacing tiles with smaller copies of the generator, while edge-replacement curves are generated by replacing the segments of a generator with smaller copies of the generator. One important consequence of node-replacement is that it requires connective segments to be added at each stage, to keep the curve unbroken, as we just saw above. Some examples of similar curves are shown below: a curve attributed to Walter Wunderlich (a); a curve attributed to Peano (b); and the Z-Order (Lebesgue) Curve (c). The Sierpinski Curve (d) is shown here because it is similar, however it is of a different class: it uses a different kind of connectivity at each stage.

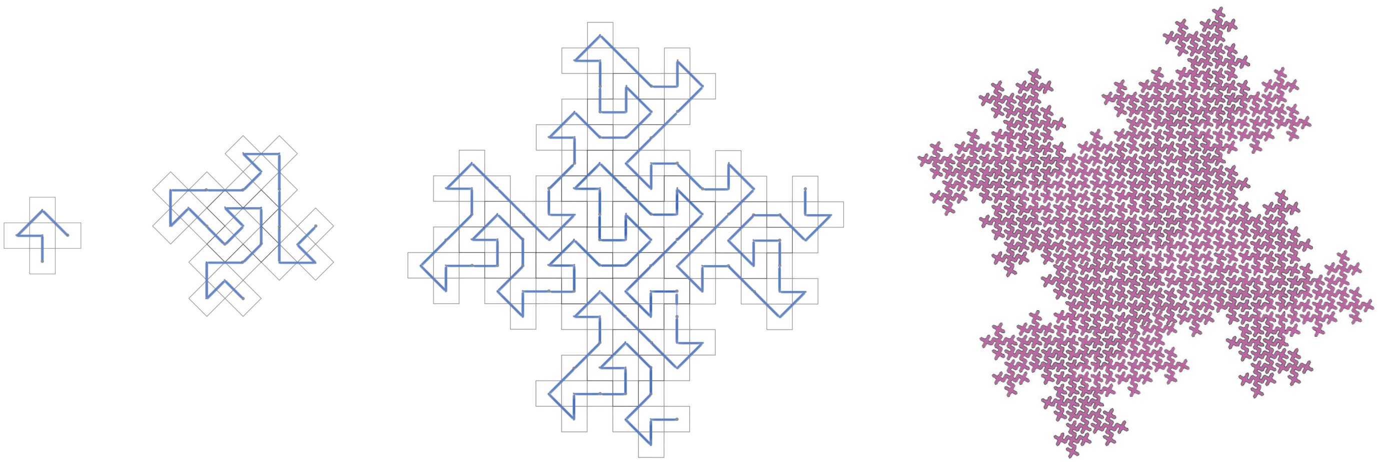


Gary Teachout [22] came up with several fractal curves using a similar technique. A few are shown here.

original shape　copies, transformed　connected　level 3

original shape　copies, transformed　connected　level 3

By the way, the resulting shape doesn’t have to be a simple polygon; it can have a fractal boundary. Here’s a generalized technique for generating curves with node-replacement: start with a collection of tiling polygons, like squares, and connect each of the tile centers with line segments, as shown below at left. Now…fractalize! Notice in this example that although there are five tiles in the initial figure, there are only four line segments. This fact is why connecting lines are required at each iteration (As we saw in the Hilbert curve, four square tiles are connected using three line segments).

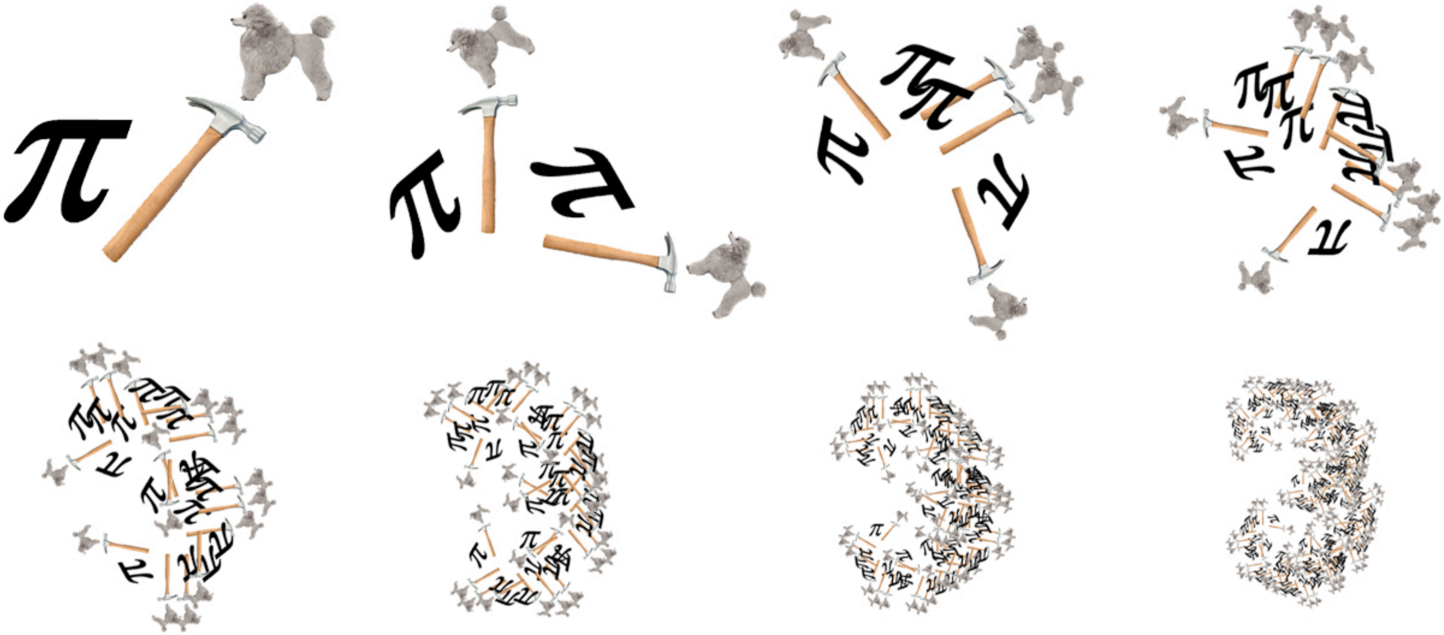


The above curve fractalizes to the same shape as a well-known fractal curve introduced by Mandelbrot (at right). I’ll be describing this curve later on.

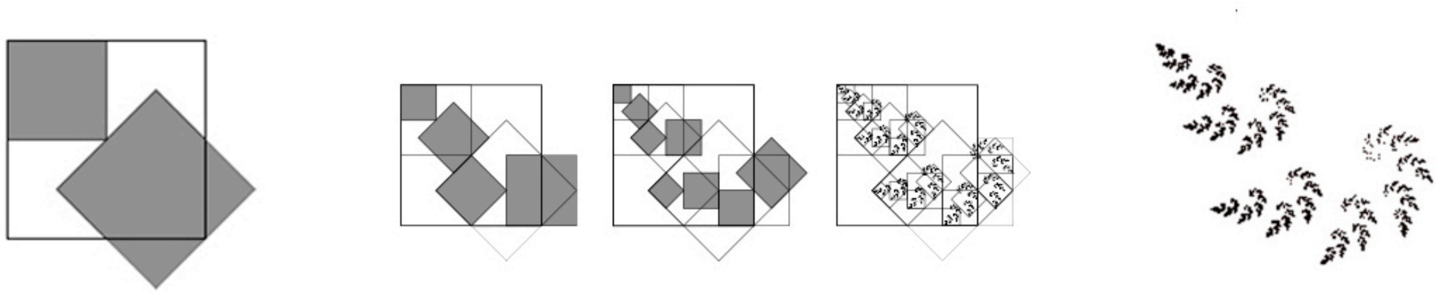
Iterated Function Systems

A more general class of fractal techniques is a process known as "iterated function systems" (IFS), conceived by John Hutchinson [10], and developed further by Michael Barnsley [2], and others. An intuitive way to describe IFS is to start with an initial image or piece of geometry, and to replace it with 2 or more copies of itself. Each copy has some transformation applied to it (e.g., rotation, translation, scaling). With IFS, copies of the original shape are progressively reduced in size repeatedly until they are essentially "atomized" as points, whose union creates the resulting shape.

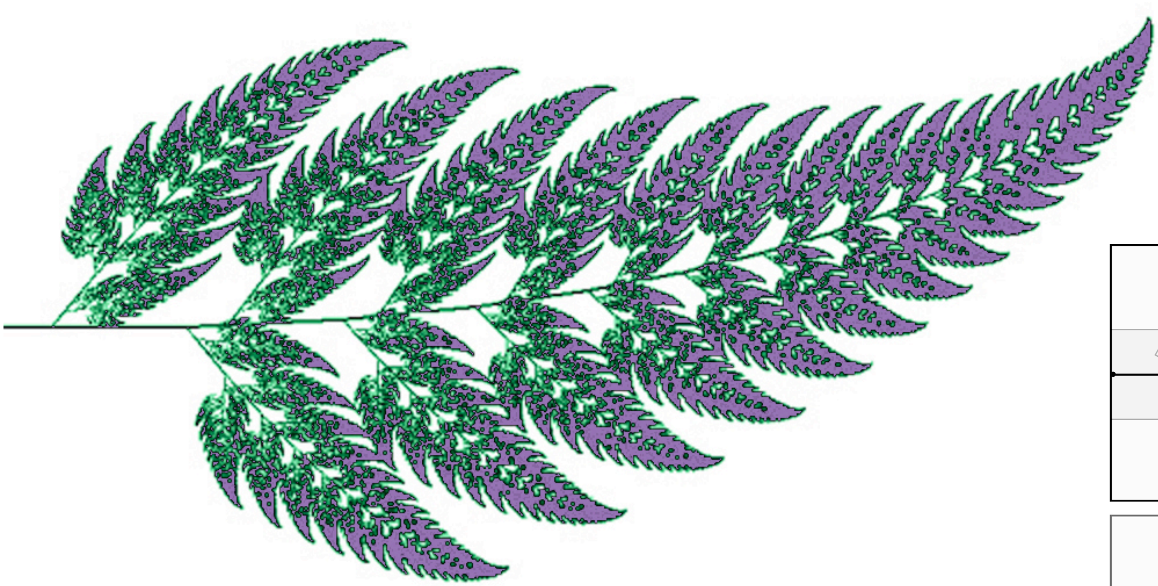
As long as the copies are scaled to be smaller than the original geometry, the ultimate result is always a "dust" – the number of visual elements goes to infinity, and their sizes become infinitely small, thus, the original geometry could be anything, basically. Take the example below of the symbol for Pi, a hammer, and a poodle. This trio of elements is copied twice, and each copy is scaled to 70% its original size. Using the bottom of the hammer as the pivot point, one copy is rotated -40 degrees, and the other is rotated 60 degrees, and also translated down and to the right a bit. All these translations happen in the frame of reference of the transformed copy. This example shows 8 iterations of this process, and you can see that a fractal has emerged with its own peculiar shape – a shape that may not have been predicted from applying the transformation only once.

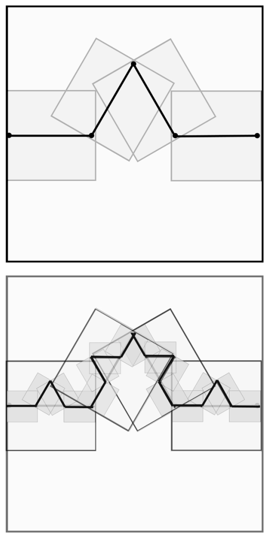


Here is another picture that illustrates this process. In this picture, the original geometry consists of two gray squares, shown at left with a larger outlined square for reference. Each gray square is replaced with the original geometry, using the same relation that the gray squares have with the larger outlined square.



The fern is a classic natural form that is often used to demonstrate IFS.

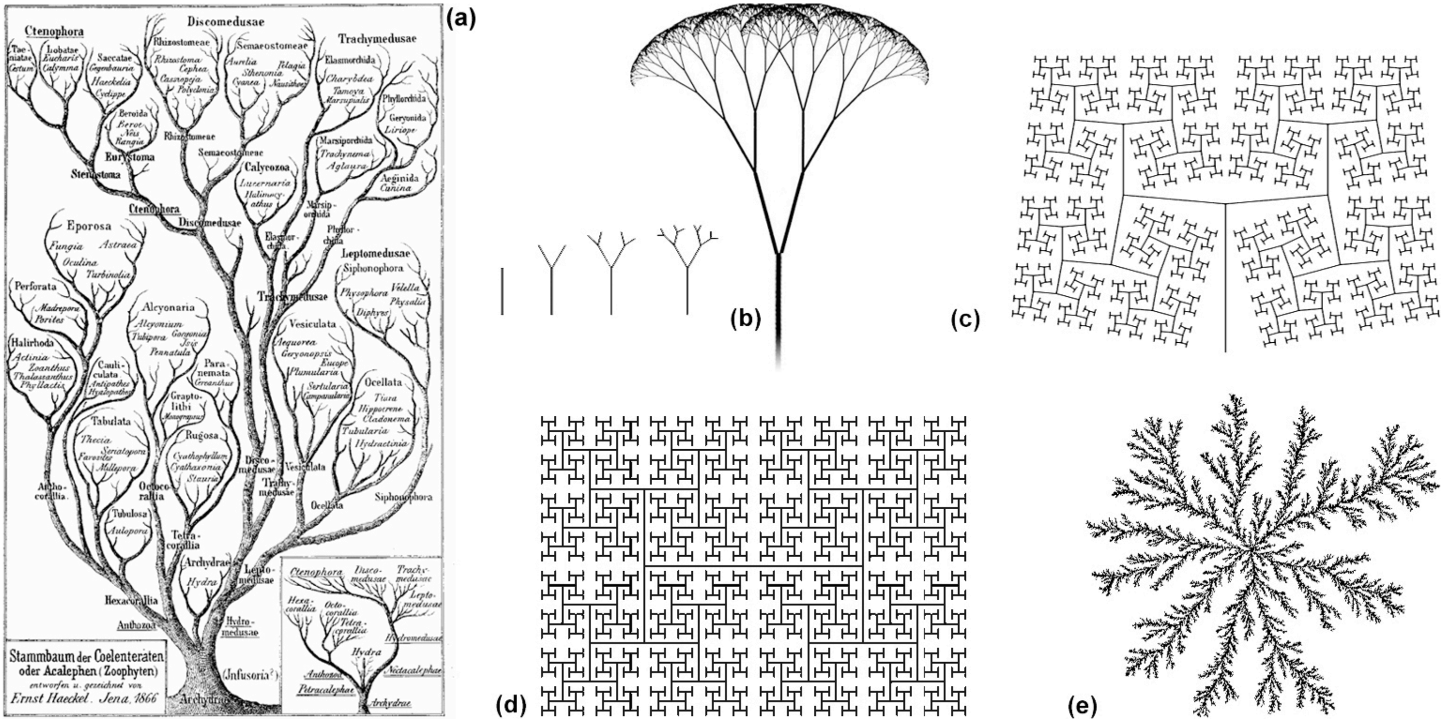




IFS can be seen as a more general way to perform Koch construction. Just consider the segments of a Koch generator as the original elements. Instead of poodles and hammers, you start with straight lines, which happen to transform at each stage so that they always form a connected chain.

Branching Fractal Trees

When visualizing hierarchical data structures with many branchings, such as classifications of biological species, one ends up with a fractal-like shape, especially if there is deep hierarchy. This is indicated below in a Tree of Life drawn in 1866 by the German biologist and illustrator Ernst Haeckel (a).

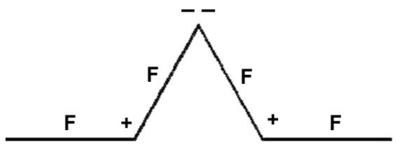


Branching fractals differ from Koch fractal curves in two primary ways: (1) they contain branch-points (obviously), and (2) they include the whole hierarchy of ancestry used to calculate each level: trunk, branch, stem, and all. Koch construction, on the other hand, *replaces* each parent generator with offspring copies when a new teragon is calculated, leaving only the smallest, most detailed segments. The classic fractal tree (a) uses a single angle at each branch point, with a uniform scaling < 1.0 of offspring segment lengths. When the angle is widened, we get a shape like (c). If the angle is 90 degrees (and if the length scaling is $1/\sqrt{2}$) it becomes the H-tree (d) – a plane-filling, self-avoiding fractal. Irregular branching forms can be created in a variety of ways, including *diffusion-limited aggregation* (e): start with a seed crystal particle surrounded by free-floating particles moving randomly. If a floating particle comes in contact with the seed, it sticks, thus extending the seed crystal, which gradually grows into a fuzzy branching fractal form.

L-Systems

In 1968, a botanist by the name of Astrid Lindenmayer devised a way of describing the growth of plants. This has come to be known as “L-systems”. They are recursive “string-rewriting systems”, used to model branching forms, embryological development, and many self-similar fractal geometries. Consider the following L-system description used to draw the Koch curve:

```
axoim:      F
constants:  + -
angle:      60
rule:       F → F+F--F+F
```

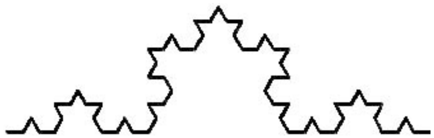


The axiom “`F`” means “draw a line segment”. The symbol “`+`” means turn left 60 degrees. And the symbol “`-` ” means turn right 60 degrees. We start with the axiom “`F`”, and we apply the rule, which replaces “`F`” with “`F+F--F+F`". This results in a teragon 1 Koch curve. Notice the use of “`--`“ to represent turning right *twice*, which makes a 120 degree right turn. Now we apply the rule again, replacing all instances of “`F`” with “`F+F--F+F`”. This gives us:

```
F+F--F+F+F+F--F+F--F+F--F+F+F+F--F+F
```

If we apply the rule a third time, replacing all instances of “`F`” with “`F+F--F+F`”, we get:

```
F+F--F+F+F+F--F+F--F+F--F+F+F+F--F+F+
F+F--F+F+F+F--F+F--F+F--F+F+F+F--F+F--
F+F--F+F+F+F--F+F--F+F--F+F+F+F--F+F+
F+F--F+F+F+F--F+F--F+F--F+F+F+F--F+F
```



Once this string has been grown, it can be given to the turtle as a single list of instructions, and the turtle goes to work.

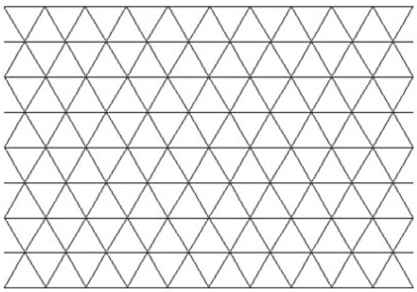
This is only the very beginning of what L-systems can do! L-systems are more than just a way to make fractals; they can be used for describing (and constructing) a huge number of forms, including Hilbert curves and its variants, branching forms, and many other shapes. L-systems can even be used for modeling embryological development over time. This flexibility is because *any* kind of alphabet can be used (not just F's and +'s and –'s) which can stand for any transformation or operation you can imagine.

L-systems provide textual representations of fractals (linear strings of alphabetical symbols that are read from start to finish). The turtle must read the entire string – like a novel or a score for a piano sonata – and follow it from start to finish. Although L-systems are elegant, terse, and very expressive, I prefer to put the turtle *into the middle of the algorithm*, so to speak, and engage the fractal growth process in a concrete, procedural way – to conceptually enter into the geometrical process of fractalization – as a visual-spatial activity. My turtle is not reading a score; it is performing a fractal dance – holding in its memory all the recursive levels and transformations required to do the dance.

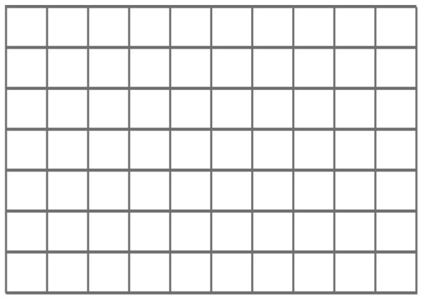
Now that I have described a few alternate techniques and variations used to make fractals, I want to now focus on the particular technique that I have devised. It is based on Koch construction, as described by Mandelbrot, but it has some new and unique differences, which I think might make it easy to understand. Most importantly, my scheme allows all Koch-constructed fractal curves to be placed into a taxonomy, allowing us to see relations and family types, and to explore variations. Now, the first thing I have to explain is the backdrop upon which all these wonderful specimens are generated: the *grid*.

Two Grids

All plane-filling fractal curves in my scheme fit into one of two kinds of grids: triangle or square. The distance between grid points is exactly 1, in either case. The most common fractal curve angles and lengths in these specimens are shown at right, associated with each grid.

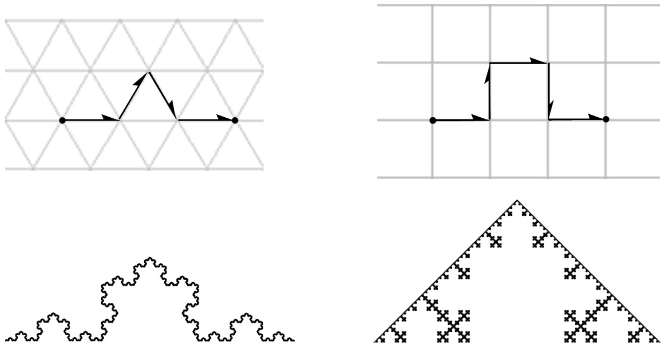


triangle grid
most common angles: ±60, 0, ±30, ±120
most common lengths: 1, $\sqrt{3}$, 2



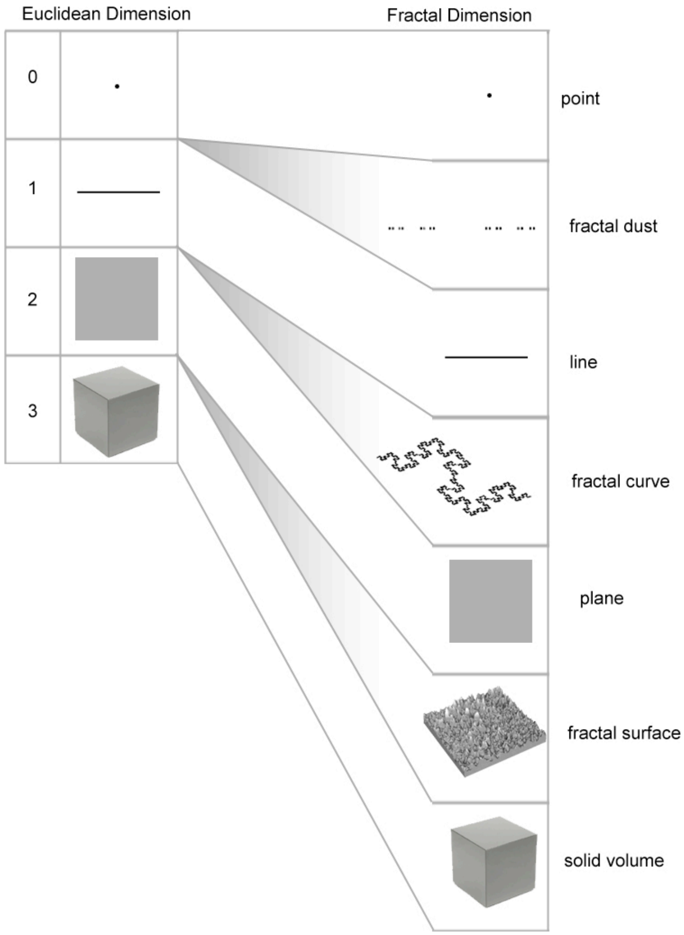
square grid
most common angles: ±90, 0, ±45
most common lengths: 1, $\sqrt{2}$, 2

Consider two curves: the Koch curve and a similar one with a square bump (sometimes called the “Square Koch”). Not only do the generators fit snuggly within their grids, but all of their fractalized teragons fit snuggly as well (that is, they fit snuggly into grids with smaller and smaller cells as fractals levels increases).



Fractal Dimension

Notice also that each of these two fractal generators covers a span of three grid units, from left to right. This span is called the “interval length” of the fractal generator. The Koch curve has four segments, but its squarish friend has five: it fills a bit more space – it’s slightly denser. More technically, it has a higher *fractal dimension*. Fractal dimension extends the idea of Euclidean dimension (integer numbers) to include fractional numbers, and lots of fuzzy structures besides.

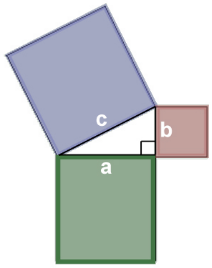


Here’s how we calculate the fractal dimension of a curve: we take the number of segments in the generator (call it **N)**, and then we take the interval length of the generator (call it **L**). Then we calculate the fractal dimension using this equation:

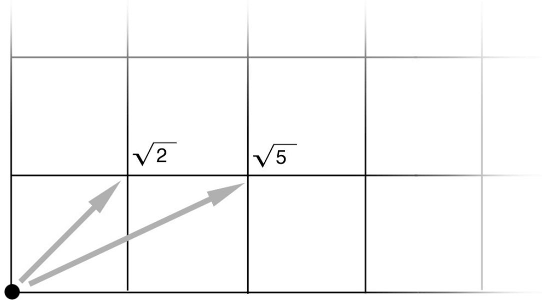
log N / log L

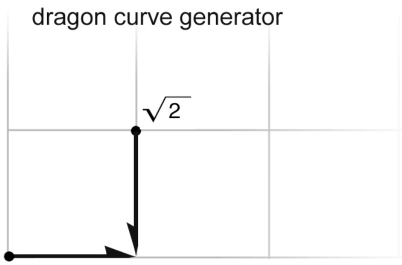
In the case of the Koch curve, N=4, and L=3, and so the fractal dimension comes out to approximately 1.2618. For the square Koch, N=5, and L=3, and it’s fractal dimension is approximately 1.4649. If the dimension of a fractal curve is 2 – and if it is well-behaved – it is a plane-filling curve.

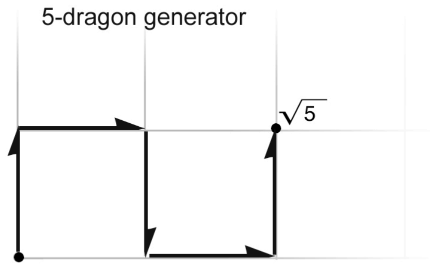
Families of fractal curves are determined according to how their generators fit within their grids, and what kinds of grids they occupy. Not all fractal generators are as simple as Koch and its square friend. And not all fractal generators rest horizontally on the grid. I'll give you two examples.



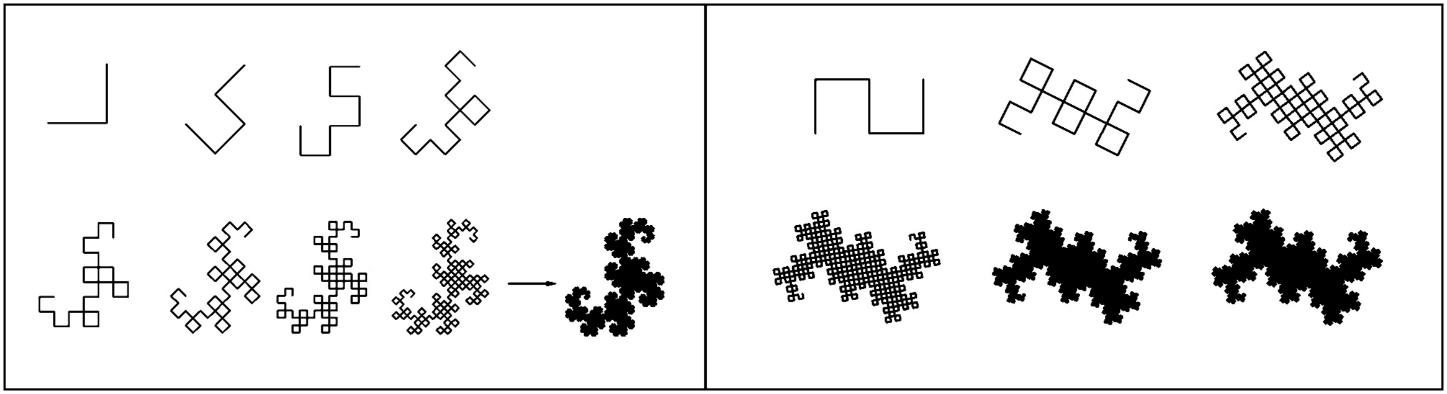
Let's draw a square grid, and then refer to a grid point in the lower left corner as the *origin*. This will be the starting point for all fractal generators. Since all plane-filling fractal generators fit into a grid, we can be sure that the end of the generator will always fall on a grid point. Below is a square grid with two of the grid points indicated. The distance from the origin to these grid points are √2 and √5, respectively. (We know this because of the Pythagorean theorem: $a^2 + b^2 = c^2$ – see diagram at right). Well, it turns out that there are two very special fractal generators that fit snuggly within these diagonal spaces. They are used to make the classic dragon curve, and the "5-dragon" (that's the name I use for a fractal I discovered many years ago).







Let's fractalize these two generators and see what happens:



What have we here? We have two plane-filling fractal curves! That means their fractal dimensions are exactly 2. And it should come as no surprise that the number of segments in each generator is exactly the square of the interval length. In other words:

$$\mathbf{N = L^2}$$

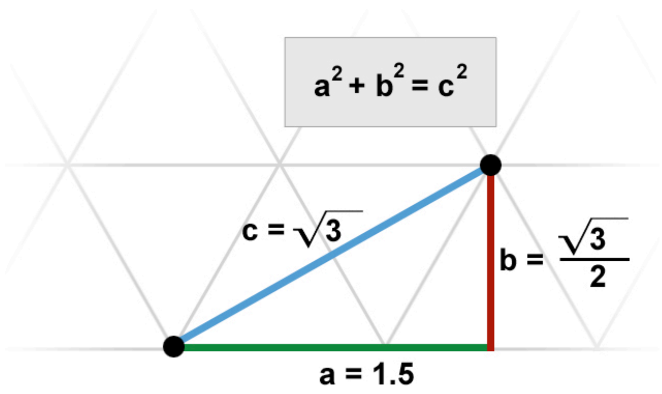
Now we have a consistent scheme for finding plane-filling fractal curves. The four fractals that I have just showed you are listed in the table below, along with their associated grid type and interval length (which I call "family type"). This table represents four different family types.

	family type		
Fractal name	Grid Type	Interval Length	Dimension
Koch Curve	**triangle**	$\sqrt{9}$ **(3)**	**~1.2618**
Square Koch	**square**	$\sqrt{9}$ **(3)**	**~1.4649**
Dragon Curve	**square**	$\sqrt{2}$	**2**
5-Dragon	**square**	$\sqrt{5}$	**2**

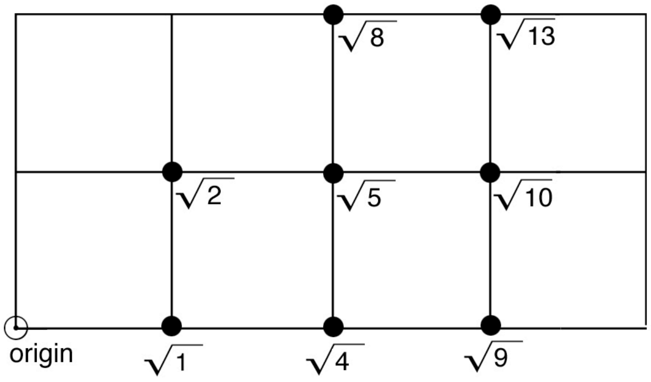
Grid Distances

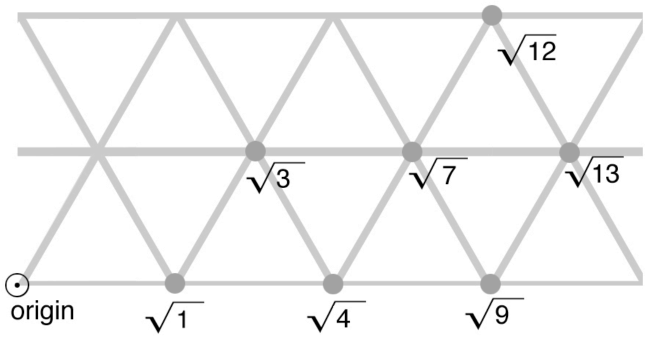
It is easy to use the Pythagorean theorem to figure out the grid point distances from the origin in the square grid, but the triangle grid is a bit more tricky. That is because the right triangle used to calculate $a^2 + b^2 = c^2$ has lengths that are not integers.

The image at right shows how the Pythagorean theorem is used on a triangle grid to find the length of hypotenuse ***c***, which represents the interval length of a fractal generator. Remember that all fractal generator intervals fall between two grid points. In the triangular grid, the length of horizontal leg ***a*** will always be a multiple of 0.5, and the length of vertical leg ***b*** will always be a multiple of $\sqrt{3}/2$. It turns out that with these length multiples, the value of ***c*** is always the square root of an integer. This is very convenient for my fractal family taxonomy scheme!



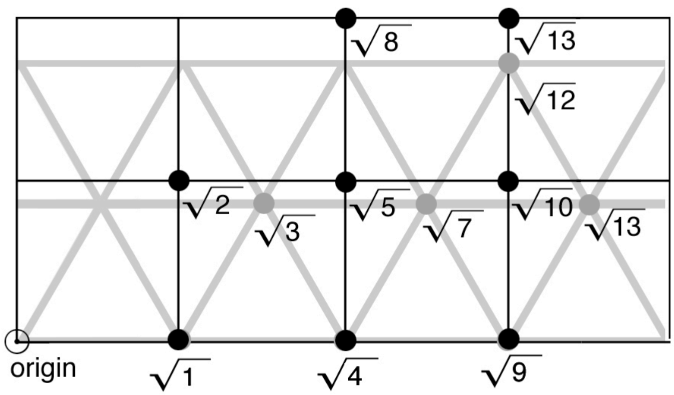
Now, look at these two grids:

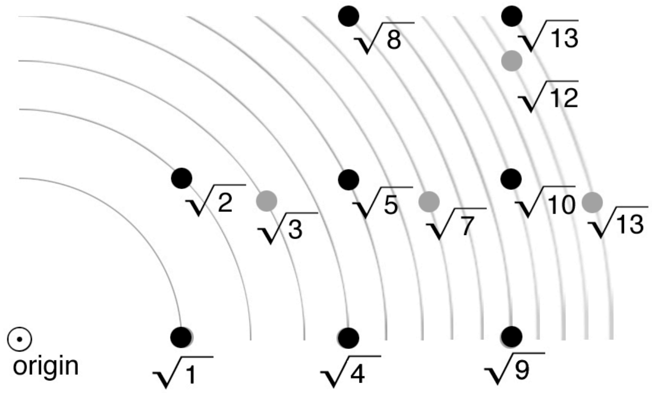




The distances from the origin to some of the grid points are shown. Notice that the distances shown at the bottom row are expressed as square roots, but this is just another way of saying 1, 2, 3.

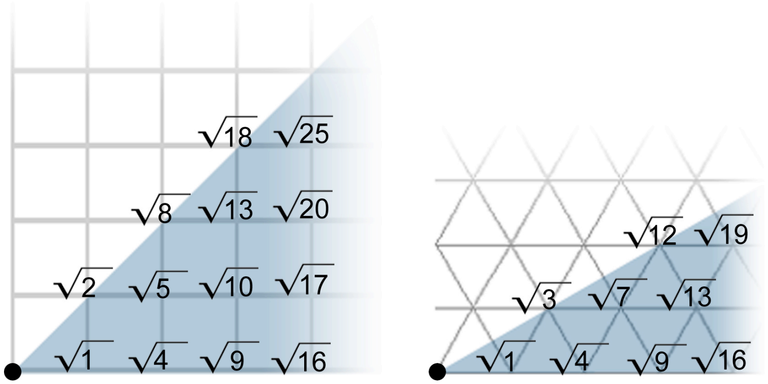
The square and triangle grids provide the backdrop for identifying all the families of plane-filling fractal curves. Let's superimpose the triangle grid onto the square grid. In doing so, we see that there are many square root distances from the origin that fall on grid points (but some are missing…like 6! – and the reason is very interesting – as I will explain later).





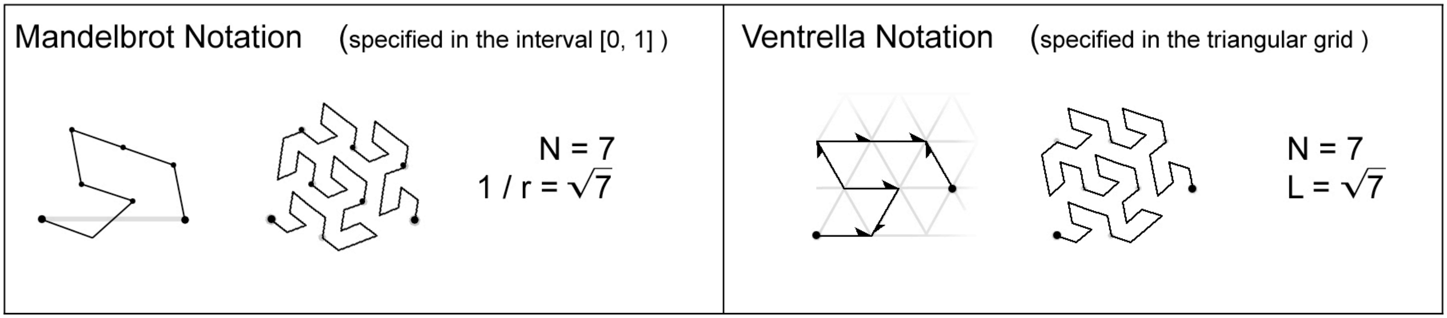
You may have noticed that I am only showing the distances within a certain clustered area in the grid. The reason is because there is no need to consider any grid points that lie outside a certain pie slice, as shown below.

For instance, in the square grid at right, an octant is highlighted, with several distances shown. All these distances could be found at grid points in the other seven octants, so we only need one octant to identify all the families. Similarly, in the triangular grid I have highlighted a pie slice occupying one twelfth the space. Any triangle-grid family type can be represented in this slice.



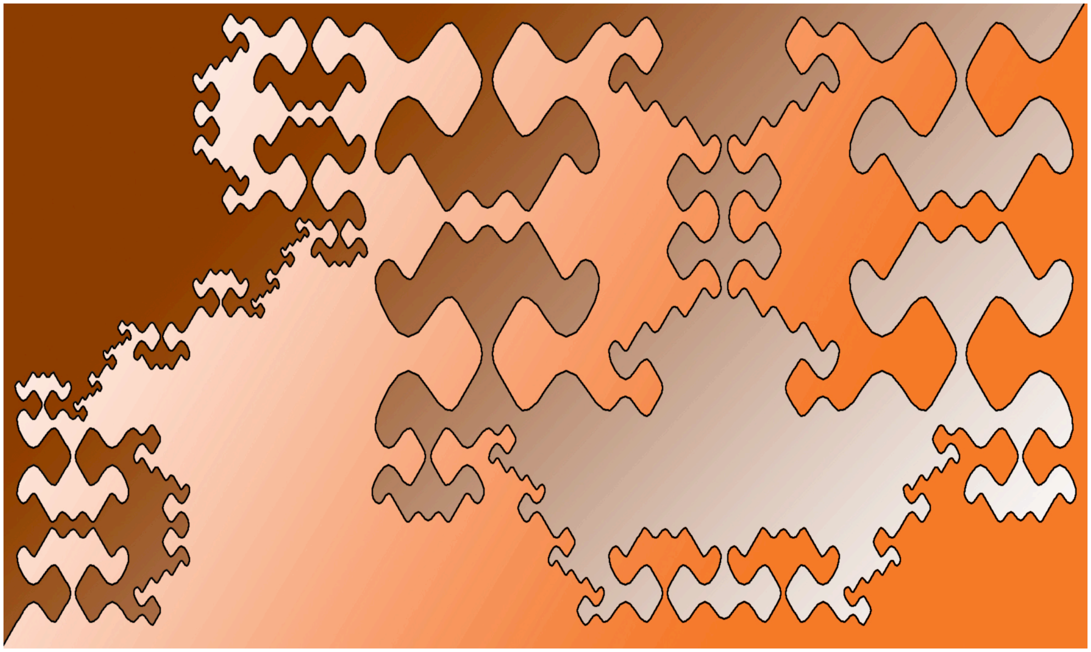
Ventrella Notation

This grid-view of fractal generators comprises the basis for how I categorize all plane-filling fractal curves. In The Fractal Geometry of Nature, Mandelbrot specifies all fractal generators as existing within a horizontal interval of one unit. The illustration below uses the *Gosper curve* as an example of how my notation scheme is slightly different than the one Mandelbrot used. Instead of specifying the generator within the unit interval such that its endpoints lie on a horizontal line, I place it within a grid, and orient it so that its second endpoint lies on the grid location associated with its family type. This causes its interval length L to traverse a portion of the grid. My notation makes it visually more apparent that the Gosper curve is a member of the √7 family type.



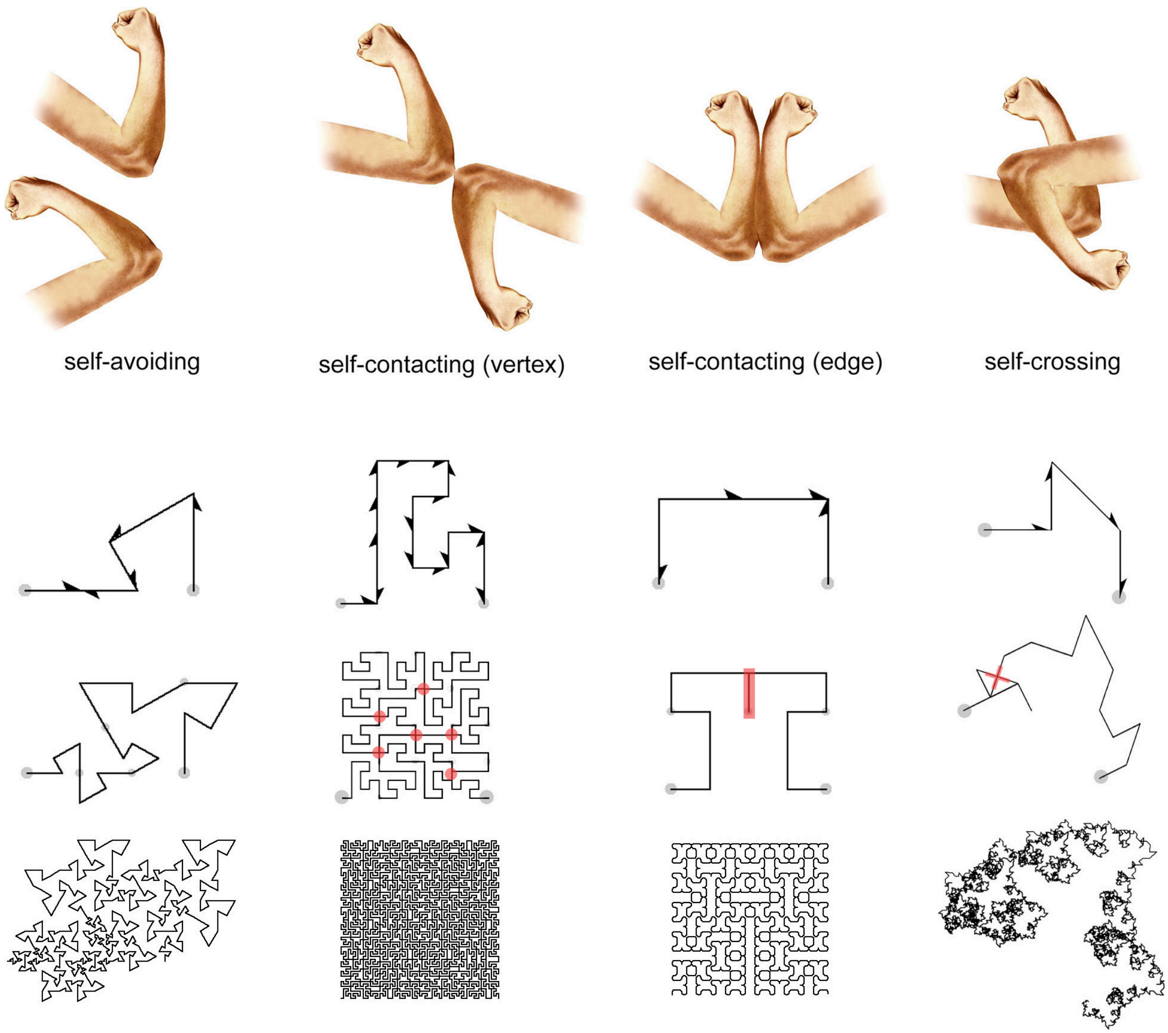
I should point out one thing about Mandelbrot's choice to place all generators on a horizontal interval of unit 1. This *normalizes* the generator so that it is easier to comprehend (and compute) the mathematical transformation of that generator into smaller copies that are then placed onto itself. The unit vector represents a normalized generator segment. My scheme may be less elegant in terms of mathematical expression, but it affords a way to classify generators within a large taxonomy, whereby the placement of the generator within the grid is the basis of the classification. Interestingly, my software algorithms necessarily transform all generators to a representation within the unit interval – for ease of computation. So perhaps Mandelbrot representation could be seen as a necessary step, both algorithmically and conceptually.

In one sense, my scheme for finding plane-filling fractal curves is simple, using the grid as a guide. However, it is not as simple as you might think (if you are novice in the fine art of fractalizing), because there are many misbehaved, self-crossing, and otherwise clumpy, hole-ridden fractals out there, obscuring the beautiful gems. And the difference between the well-behaved plane-fillers and the misbehaved ones is not something you can easily determine just by looking at a generator. For many generators, especially the ones with large interval lengths, you just have to fractalize them and find out. And you have to be patient.



Self-avoiders and Gridfillers

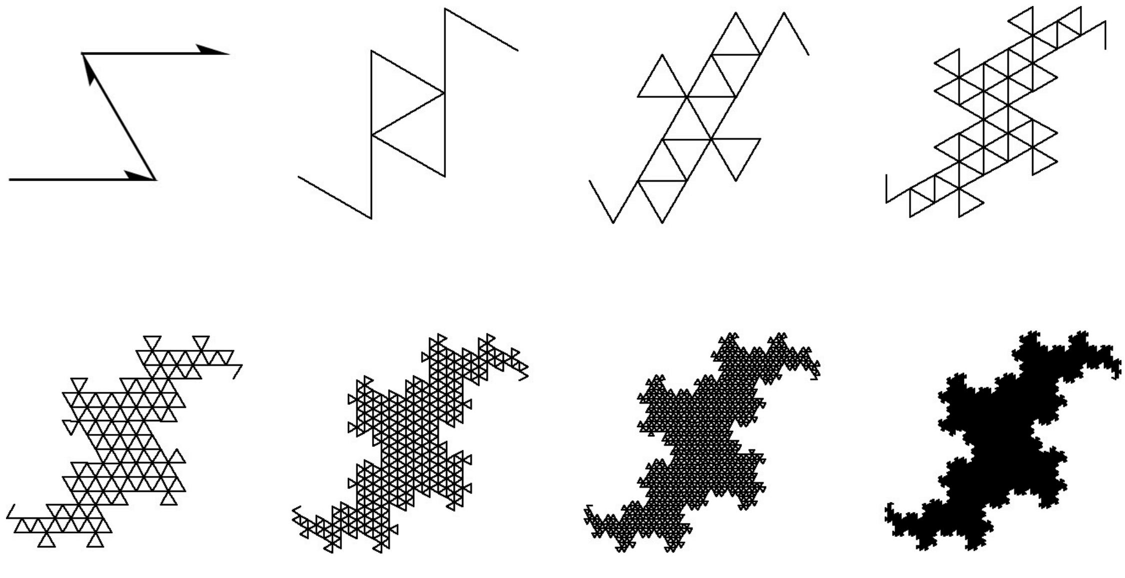
There are a few more points I want to make before we take the tour of the specimens. I mentioned that some fractal curves are "well-behaved", but most are not. I need to be a little more specific than that. There are four distinct ways that a fractal curve can behave, in terms of how it interacts with itself:



Because of the recursive nature of fractalization, any self-touching or self-crossing that occurs is sure to propagate with each fractalization…approaching infinity at higher levels. But self-touching and self-crossing fractal curves are not necessarily bad. In fact, some of the most interesting-looking fractal curves cross-over themselves excessively. And if you are a fractal explorer, I encourage you to fish for whatever strikes your fancy. But there is an intellectual – and in a certain way, aesthetic – satisfaction in finding fractal curves that are well-behaved. Also, *finding them* is a challenge – and we do so love a challenge! So, the majority of my specimens are either self-avoiding or self-touching on their vertices.

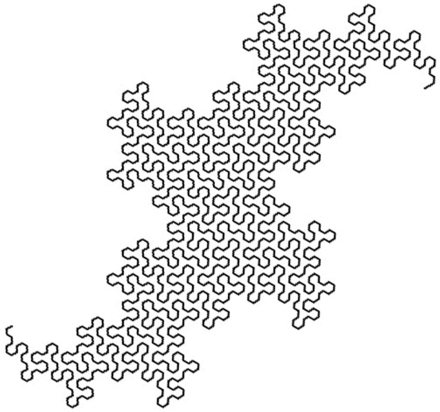
The most exciting discoveries are the "***FASS*** curves". "FASS" stands for space-**F**illing, self-**A**voiding, **S**imple, and self-**S**imilar. I have already defined "Space-filling" and "self-avoiding". "Simple" means the curve does not cross itself, and "self-similar" means that the curve appears the same at different magnifications and rotations. The recursive operations used in Koch Construction, L-systems, and IFS insure self-similarity.

Now, there is a certain class of vertex self-touching I want to mention, in which the fractal curve touches itself at *every* grid point within the area that it covers (except for the boundary). These are what I call the *gridfillers*. The dragon curve and the 5-dragon that I showed you earlier are both gridfillers. They fill a portion of the square grid. They can be described as "chunks of lattice" (in Mandelbrot's words). There are also many fractal curves that fill triangular grids, such as the Ter-dragon (shown below, with its first eight teragons). It will be described in more detail later on.

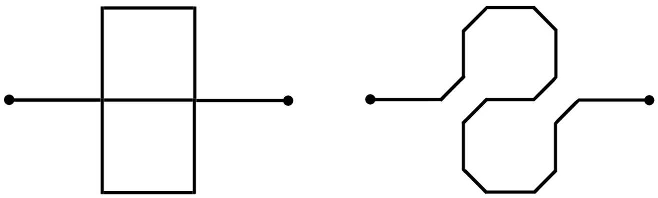


Rounded Corners

Since the Ter-dragon is a "chunk of lattice", it is hard to see how the curve sweeps through its body. To reveal the sweep of the curve, we can use *rounded corners* (more specifically: *beveled*, or *chopped-off,* corners). At the right is the Ter-dragon's 6^{th} teragon, rendered with rounded corners.

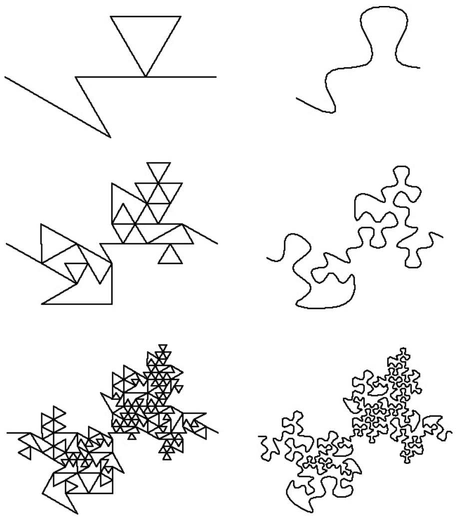


For a triangle gridfiller, chopping off the corners makes them hexagon-like rather than triangle-like. The 9-segment generator shown below lives in the square grid; chopping off its corners makes them octagon-like rather than square-like.



You will notice that many of the specimens in this book have curly lines, and that the curls appear to be self-avoiding. That is because I have rendered them with rounded corners. So, don't get confused if you think you are looking at a specimen that is a self-avoider! To make sure it is clear, the diagrams always specify if a teragon is drawn with rounded corners.

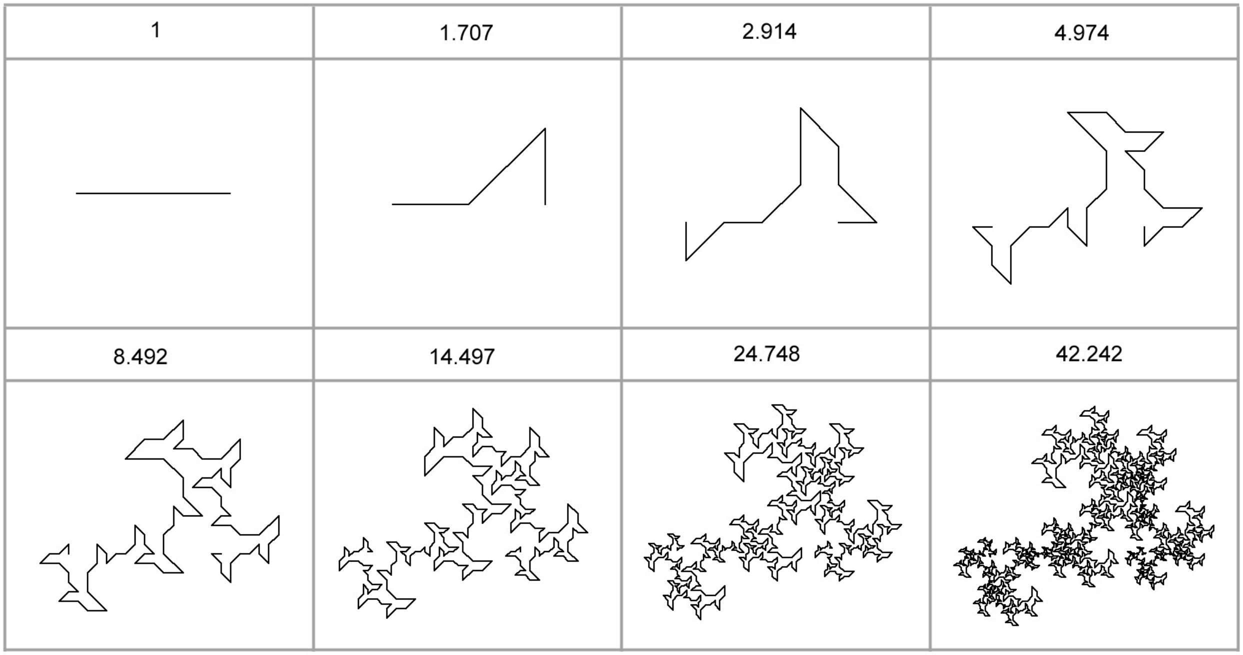
Splines

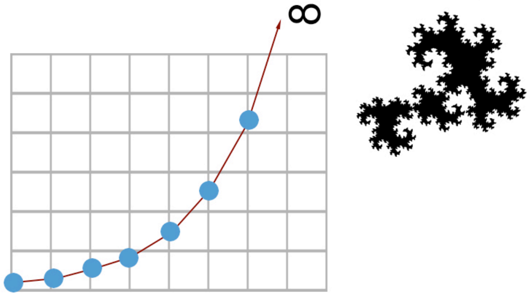


Some of the high-resolution renderings of curves in this book employ splines, which are even smoother than the chopped-off corners used in the diagrams. The picture at right shows an example of a fractal curve whose sharp corners normally touch each other. Splines help to separate these touch points, and they also give the curve a smooth organic contour.

How Long is a Fractal Curve?

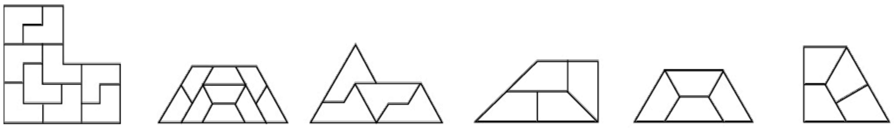
Good question! Here's the answer: *Infinite*. And here's why: whenever you fractalize a teragon, you multiply its overall length by some constant value which is greater than 1. You may have heard Mandelbrot's famous question, "How long is the coast of Britain?" Well, it depends on whether you are measuring it with the flight of a jet plane, the mileage recorded from a car ride, or the meandering of a crab crawling along the shore. Since a true Platonic fractal has an infinite number of fractal levels of self-similarity, the length essentially becomes infinitely long (even though it still occupies a finite area) This is illustrated below with the Dragon of Eve. This fractal curve's length is multiplied by ~1.707 at each level.

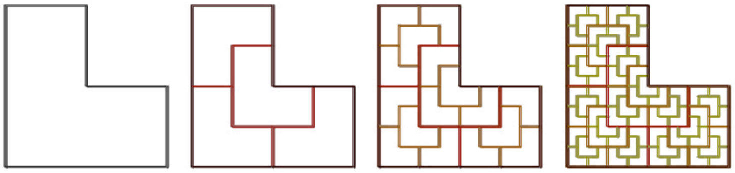


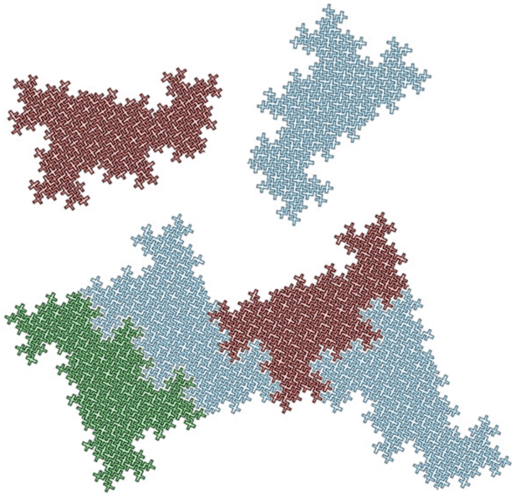


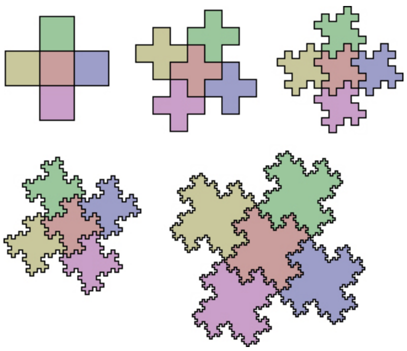
Per-Tiling, Rep-Tiling, and Penrose Tiling

Since a fractal curve is made of smaller copies of itself, it logically follows that a plane-filling fractal curve is a filled-in shape that is made of smaller filled-in shapes – identical to itself. This means that plane-filling fractal curves are *tiling*. Not only are they tiling, but they are *recursively tiling*. Mandelbrot called this "pertiling" (using the prefix "per" as in *perfume*: to thoroughly fill with fumes). These tiles are also examples of "rep-tiles". A rep-tile is a plane figure that tiles the plane and can be divided into several smaller copies of itself. Some examples are shown below at left. At right are pertilings (fractal rep-tiles) of a specimen I will show you later.

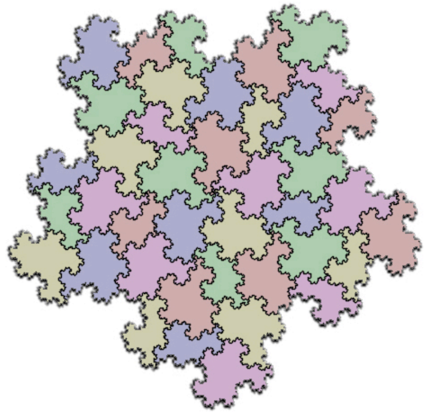




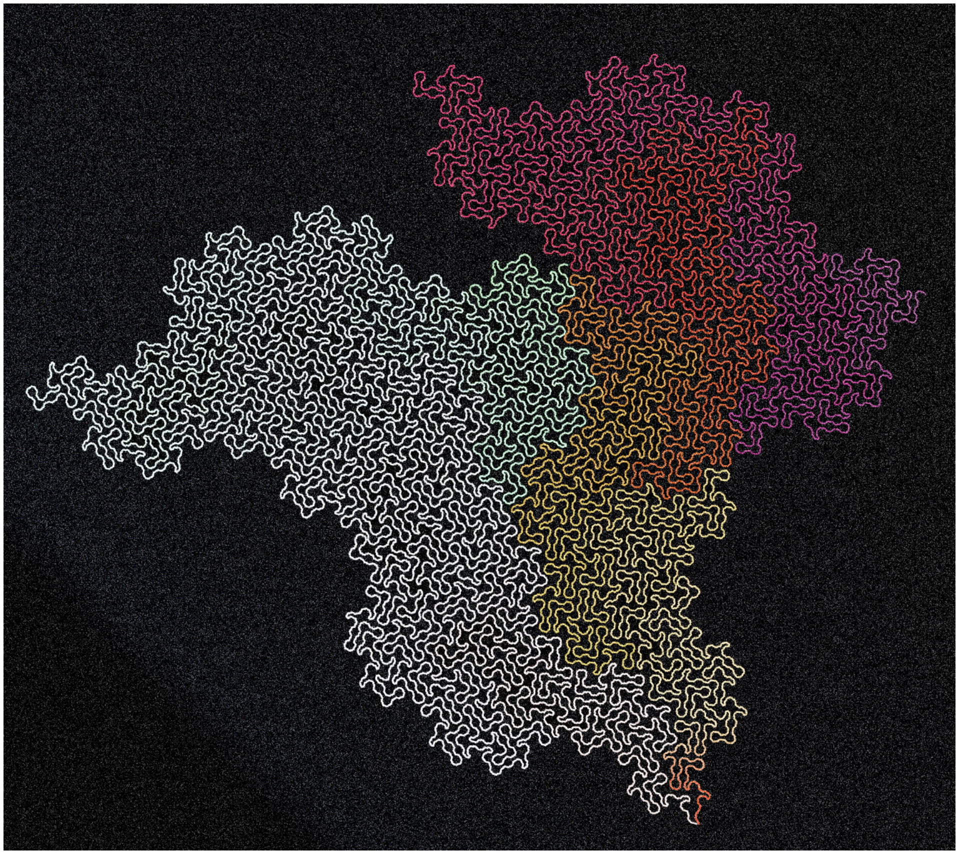




Some fractal explorers, such as Tom Karzes [11], have designed pertilings that are *a-periodic*, called "Penrose Fractals". This means these tilings can extend out forever and never repeat the same pattern. One of Tom's designs is shown at right. Penrose Fractals are further described by Bandt and Gummelt [1], and Gelbrich [8].



When I put two related fractal specimens together like puzzle pieces, I like to use the term *pertiling* – especially if they have wild fractal boundaries. Sometimes I refer to this as “mating”. Why? Because only specimens of similar species can fit together. You’ll see as we meet the different families of plane-filling curves that each family has its own particular type of morphology. Below are two related specimens mating. The ends of their curves touch at the bottom of the image.



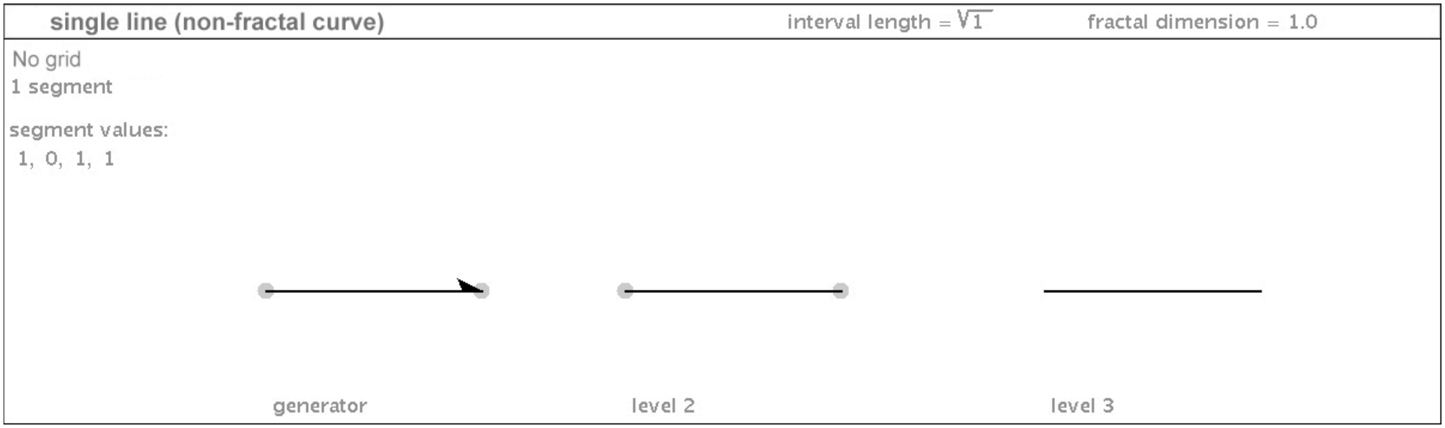
Okay, I believe we have covered enough of the basics now. In the pages that follow, I will be showing you more than 200 specimens of plane-filling curves. I have decided to leave some of the remaining interesting concepts for later, as they come up in relation to the various specimens. There’s a lot more to look at and a lot more to think about.

Let the fractalizing begin!

4

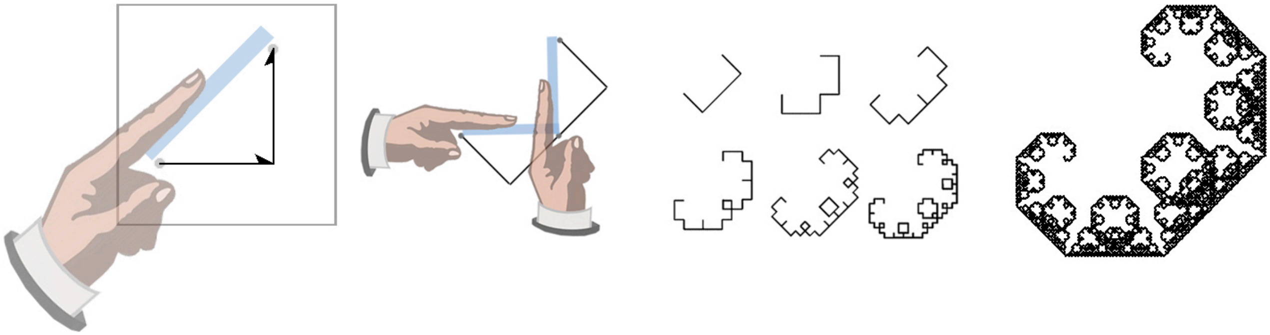
Gallery of Specimens

For the sake of completeness I will start with the simplest specimen of all: a *non-fractal* curve consisting of a straight line. Its fractal dimension is 1, and its interval length is 1. I will use this as an introduction to the diagrams used throughout the book. In this diagram, the header bar at the top shows the name of the fractal at the left (although many don't have names). To the right of that are the interval length (expressed as a square root) and the fractal dimension. Below the header bar at the left is some information about the genetics of the fractal generator. This includes the grid type (not relevant in the case of this single line), and the number of segments. Below that is a list of numbers that specify the segments in the generator. Each line segment in the generator is specified using four numbers. The first two numbers specify its displacement within the grid. In this example, the line extends one unit in the x direction and 0 units in the y direction, and so the numbers are 1 and 0. The third and fourth numbers describe the segment's flippings. I will explain that next.

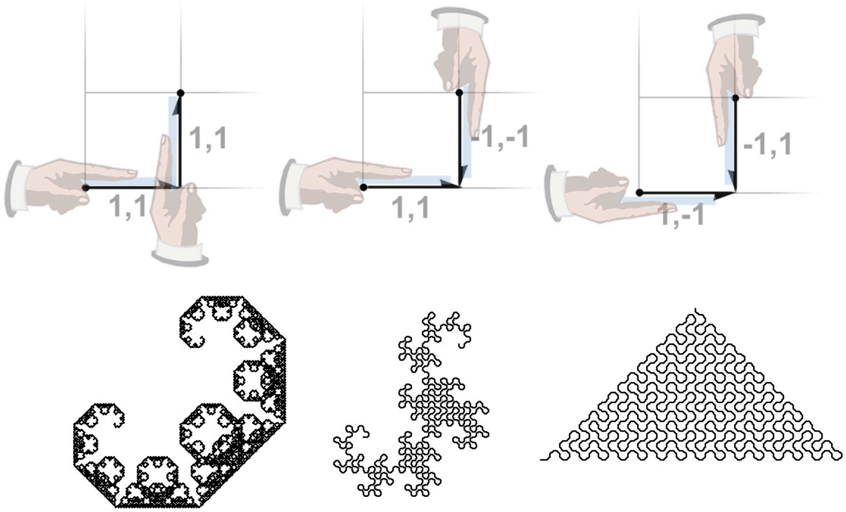


Segment Flippings

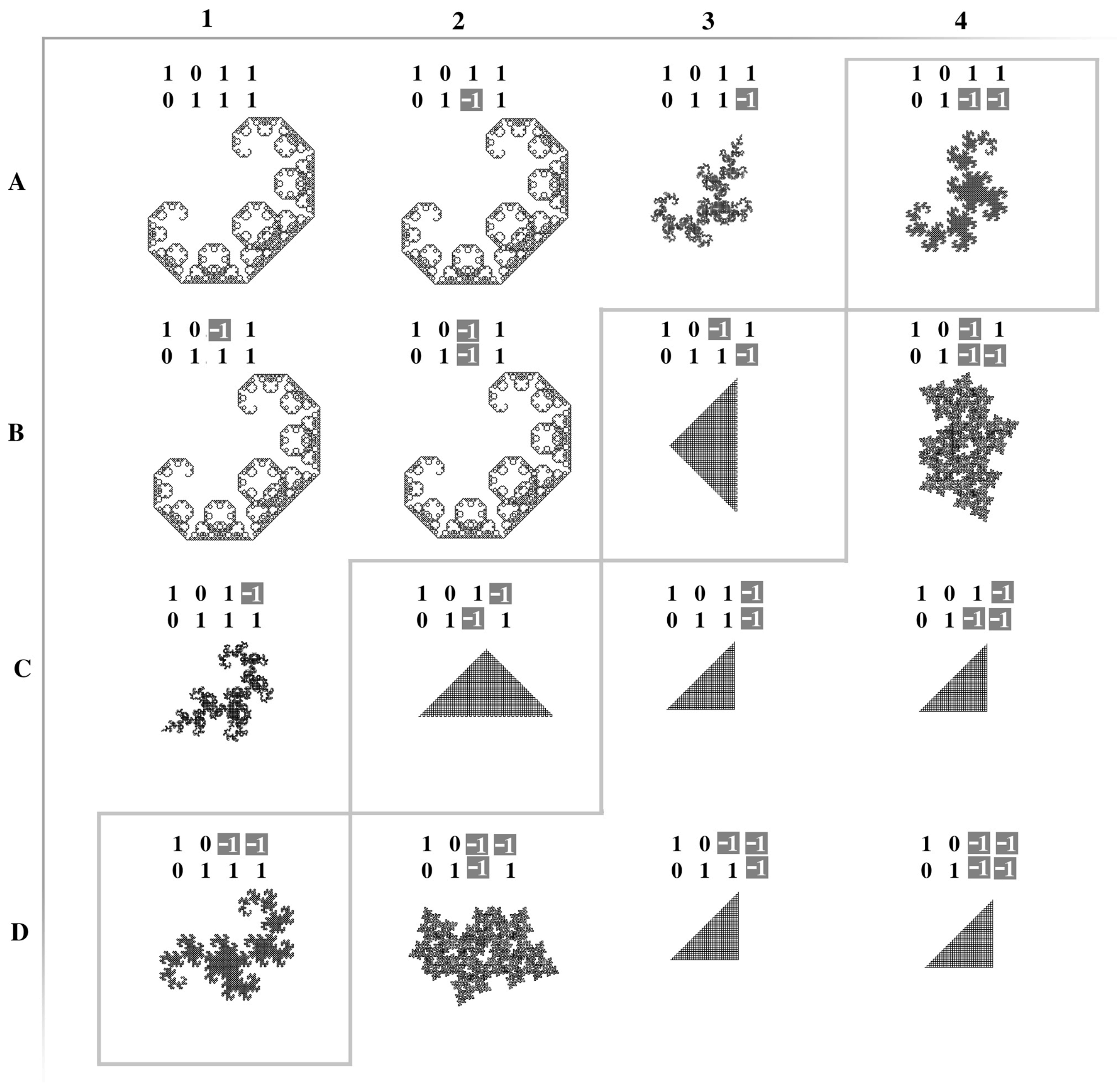
Remember the four flipped variations of the turtle I showed you earlier? These four kinds of flippings are represented in the third and fourth numbers. So: 1, 1 means no flipping; -1, 1, means it is flipped in x; 1, -1 means it is flipped in y; and -1, -1 means it is flipped in both x and y. Now let's look at an L-shaped generator with no flippings. It creates a fractal known as the Lévy C-curve:



This in an interesting fractal curve – in a gnarly kind of way. Now, consider what happens when we try a few different flippings among these two segments. Take note of the subtle difference in flippings here:



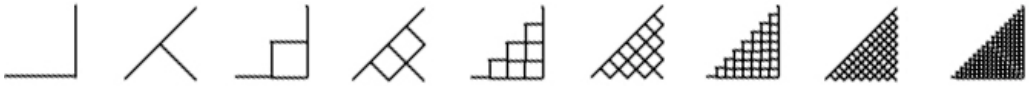
Quite different results, eh? There are in fact 16 different possible ways to flip these two segments (since each of the two segment can be flipped four ways: $4^2 = 16$). Here are the fractal curves that result from all possible flippings:



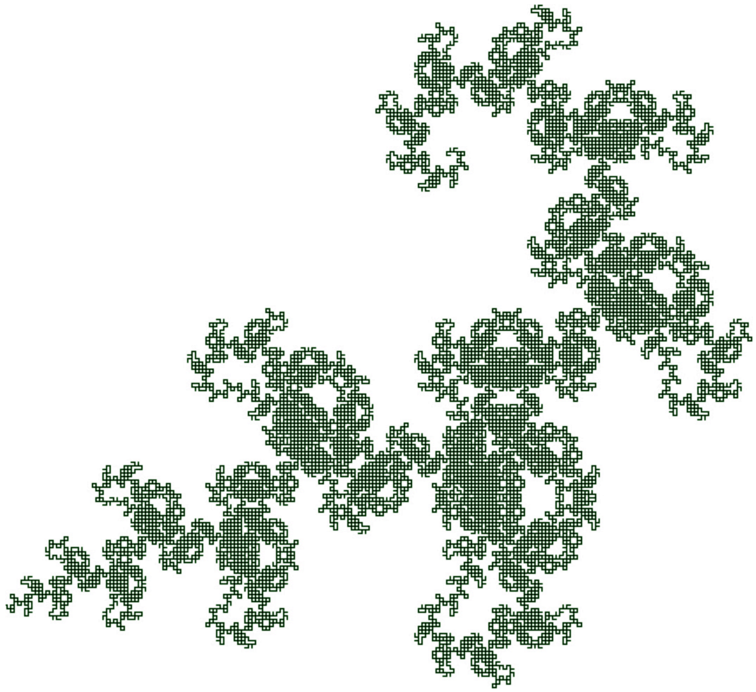
In the graph, I have labeled the rows A, B, C, D, and the columns 1, 2, 3, 4. Notice the diagonal symmetry mirrored along the axis that stretches from top-left to lower-right (A1, B2, C3, D4). Also notice the four boxes arranged along the opposite

diagonal (A4, B3, C2, D1). They specify the only well-behaved fractal curves of this family. And they happen to be gridfillers. You can see that the well-behaved fractal curves come in two forms (which I will introduce shortly). Two of them are simply flipped versions of the other two, and so we conclude that there are really just two plane-filling curves of this family, which I call the √2 family.

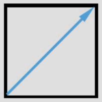
The four flippings in the upper-left corner all result in the Lévy C-curve. And the four curves in the lower right corner all result in Cesàro's Sweep, which is a *double density gridfiller*, meaning, it is everywhere self-touching along its edges. Here is a diagram showing the fractalization of the L-shaped generator to create Cesàro's Sweep:



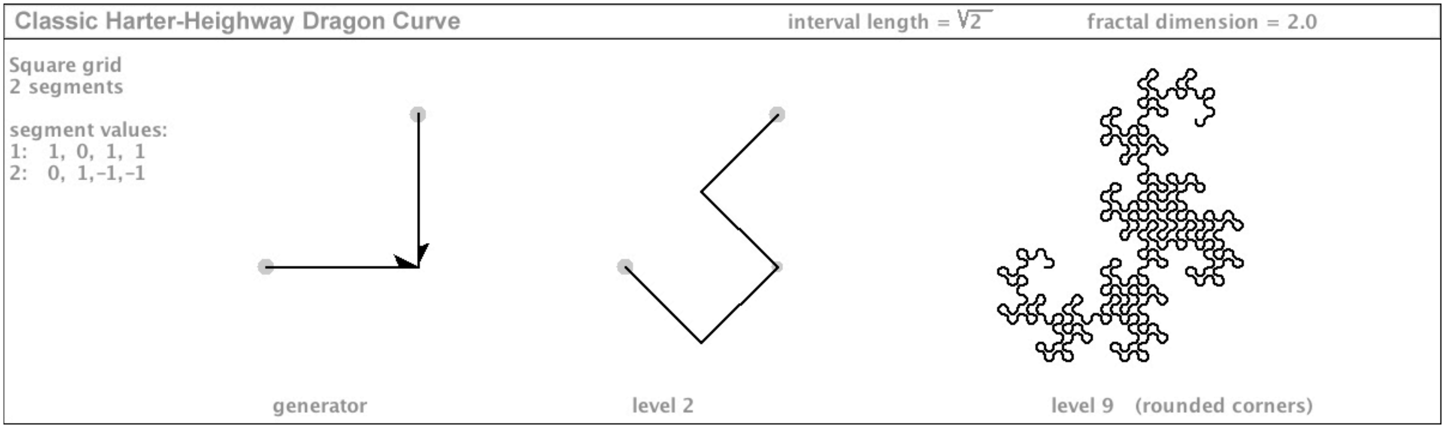
The fractal curves located at A3, B4, C1, and D2 are quite misbehaved: they cross over themselves and they leave lots of holes in the process. This is not to say that they are uninteresting. In fact, as a nod to all the misbehaved fractal curves in the world (which is most of them) I shall offer a portrait of the fractal curve at C1…here.

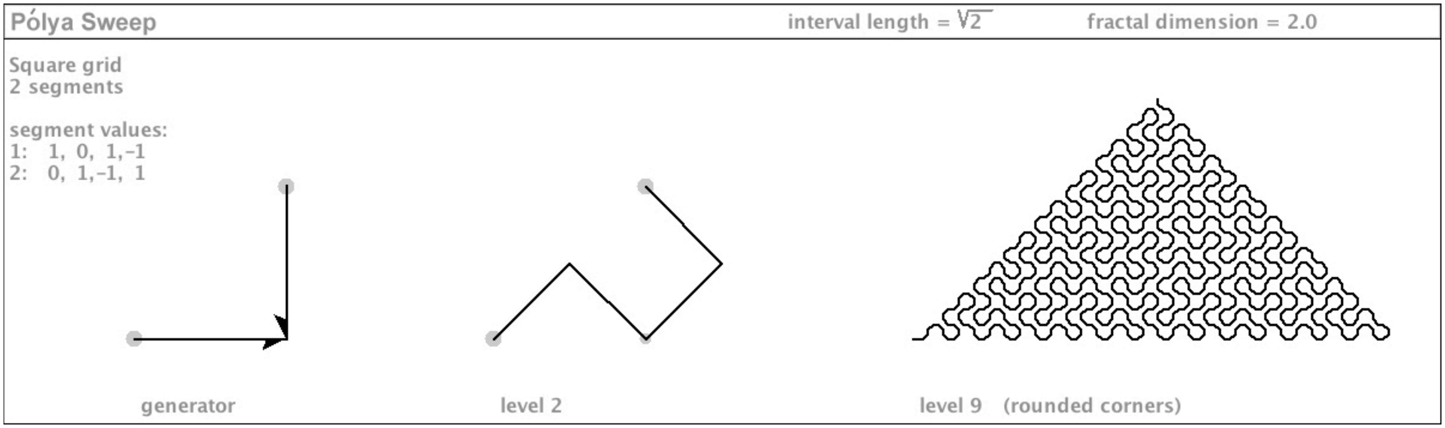


The fractal curves that self-cross or self-touch can be considered as creatures that have reinforced regions in their bodies. The density of the fabric of their flesh is uneven – some spots are thick – other spots have holes. Although they may lack the aesthetic elegance of plane-filling curves, they often do exhibit some interesting forms of self-similarity, and they evoke familiar forms in nature.

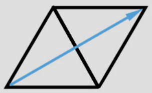
√2 

It is time to officially introduce the first family of fractal curves: the √2 family. The first specimen I will show you is perhaps the most famous one of all: the classic dragon curve. Mandelbrot called it the “Harter-Heighway Dragon”, after two mathematicians who explored it. It is sometimes referred to as the “Jurassic Park Dragon”. For convenience, I will just call it the “HH Dragon”. The other plane-filling curve of the √2 family is the Pólya Sweep. Such a very different fractal! In fact, if we continue to fractalize these two curves to high-level teragons, the HH Dragon becomes increasingly craggy along its boundary, while the Pólya Sweep becomes increasingly straight along its three sides. What’s up with that?

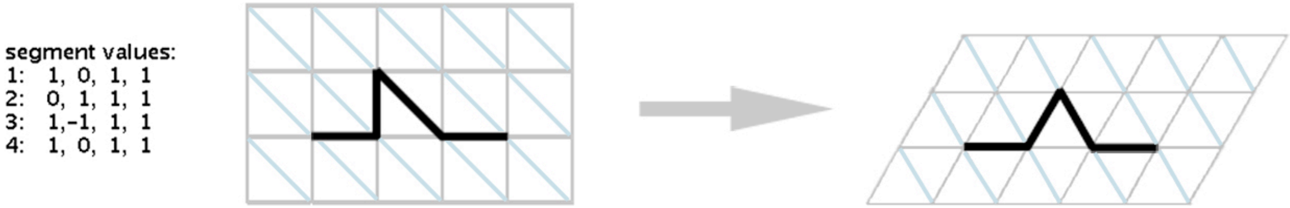




√3

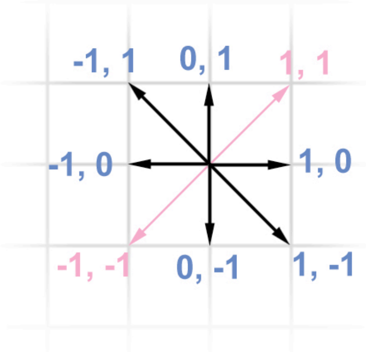


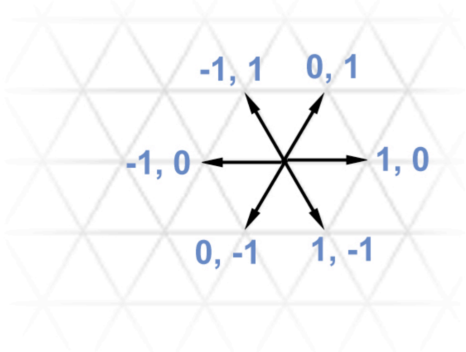
The next family of plane-filling fractal curves is the √3 family. Unlike the fractal curves of the √2 family, these curves live in the triangular grid. Now I need to explain one aspect of my notation for fractal generators in the triangular grid. Look at the illustration below:



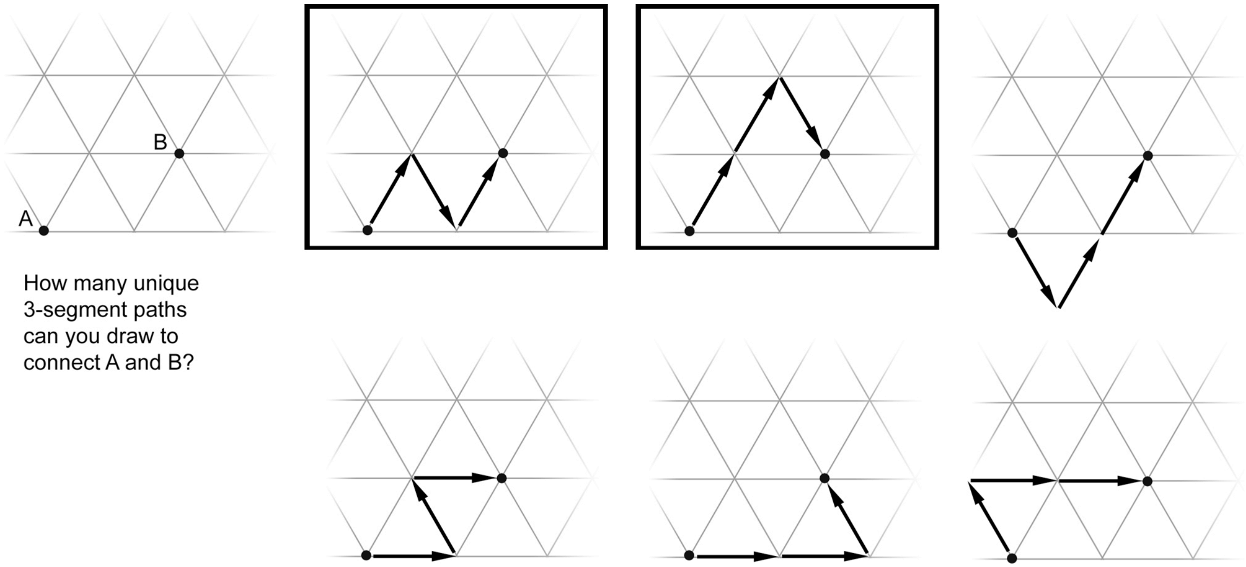
The segment values at left specify the directions that the four Koch curve segments are drawn. These values are pretty straightforward when applied to a square grid, but not so much when applied to a triangular grid. In order to use a consistent scheme for specifying segment directions in both girds, I specify all values *as if* they were in the square grid. So, for instance, the Koch curve is shown above with the following values: (1, 0) (0, 1) (1, -1) (1, 0). Since the Koch curve is a member of a triangular grid family, all of its segments have two transforms applied to them: a *skew* (notice that the top of the graph is shifted to right) and a *scale* (notice that the height of graph is squashed). These transformations result in grid lines that cross at 120 and 60 degree angles instead of at 90 and 45 degree angles.

Please keep this in mind whenever you are reading the segment values for generators in the triangular grid. To make it easier, I have drawn another picture at right. It shows six direction values (colored blue), which are mapped from the square grid to the triangular grid.





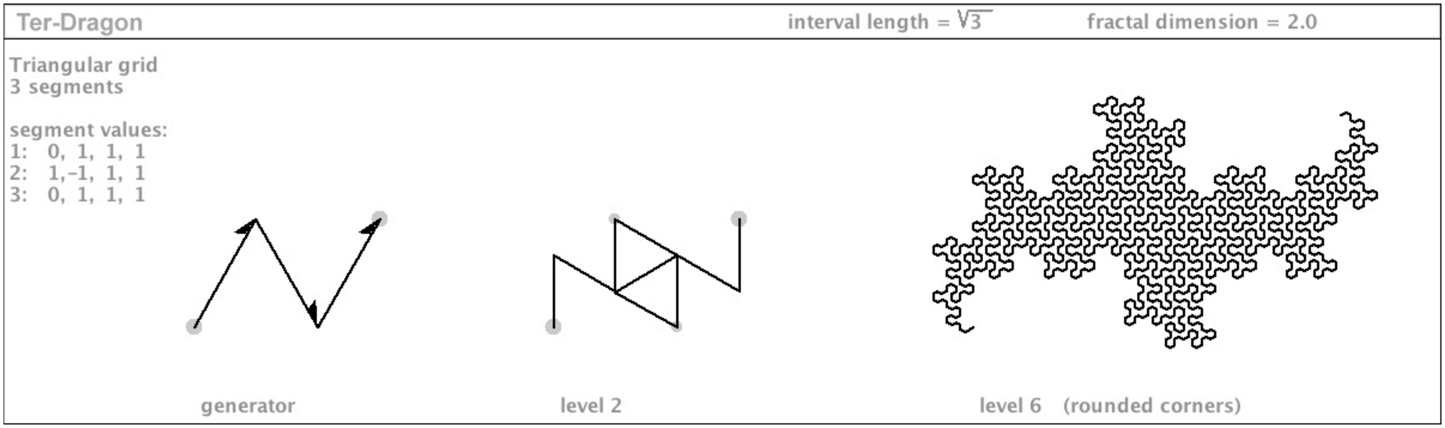
Here is a question regarding the triangular grid: look at the picture below. How many unique 3-segment paths can you draw from point A to point B?



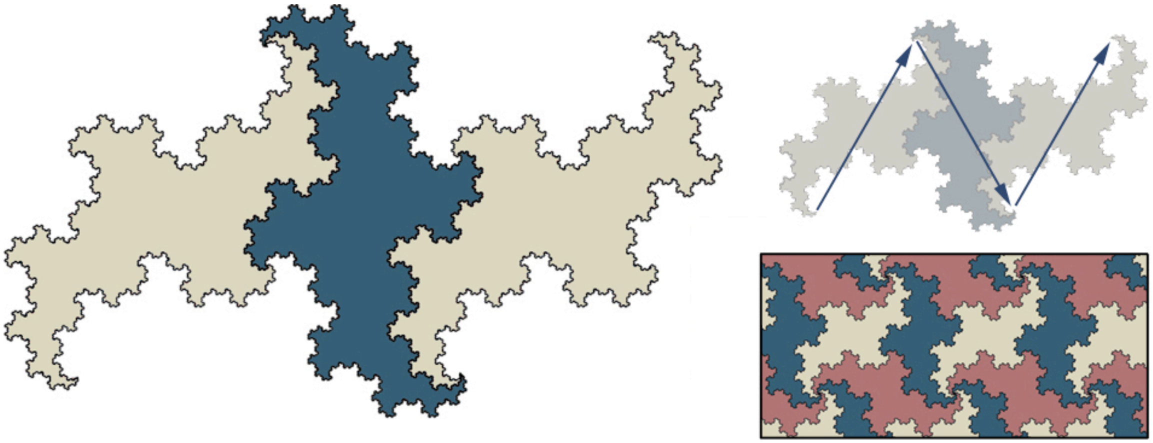
As you can see, there are six ways to connect A to B using 3 segments of length 1. But in fact there are really only two, if you consider the fact that all of them are just rotations or reflections of the two examples highlighted at the top. We can ignore the rest of them because, as fractal generators, their teragons look exactly the same as the teragons of these two, just that they are rotated or reflected. Now, given these two generator shapes, consider the various ways that each of their segments can be flipped: since each segment has exactly four possible kinds of flippings, we can conclude that the number of possible flipped variations of each of these paths is:

$$4^3 = 64.$$

Thus, there are 128 curves to test for plane-filling (64 for each of the two paths shown in the illustration). I have tested these and have found that there are ten plane-filling curves of the √3 family. One of them may already be familiar to fractal fans: the Ter-dragon:

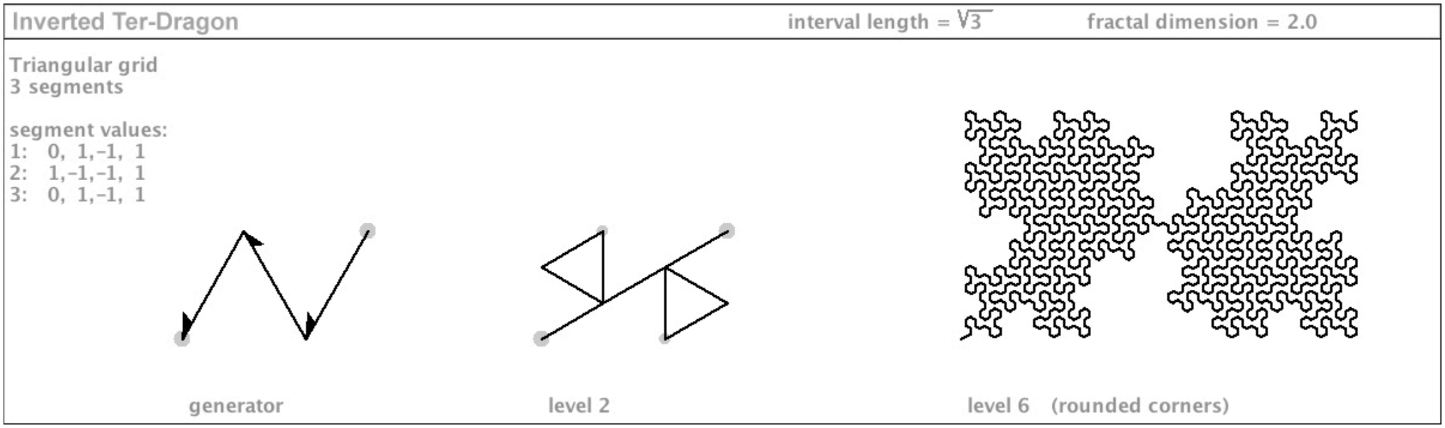


Famed computer scientist Donald Knuth is said to have first discovered the Ter-Dragon. Unlike the HH Dragon, the Ter-Dragon has point-symmetry: its tail looks like its head…which looks like its tail. Notice also that three copies of the Ter Dragon can be combined to make a larger one. But no surprise there, right? This fractal curve just oozes with three-ness. The box at the lower-right shows how the Ter-Dragon can tile the plane.

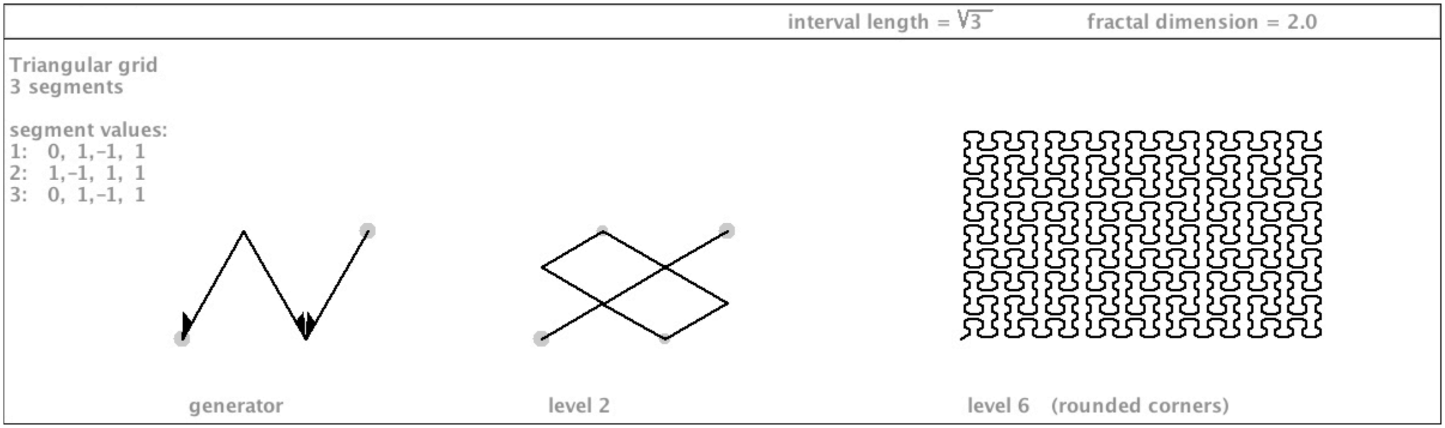


The Ter-Dragon is our first example of a "Palindrome Curve", that is, a fractal curve which is symmetrical about its center. Palindrome Dragons have heads that look like upside-down copies of their tails. I'll be showing you more interesting properties of Palindrome fractals later on.

Based on the Ter Dragon's generator, we can create an entirely different palindrome curve simply by flipping each of the segment x values, as shown here:

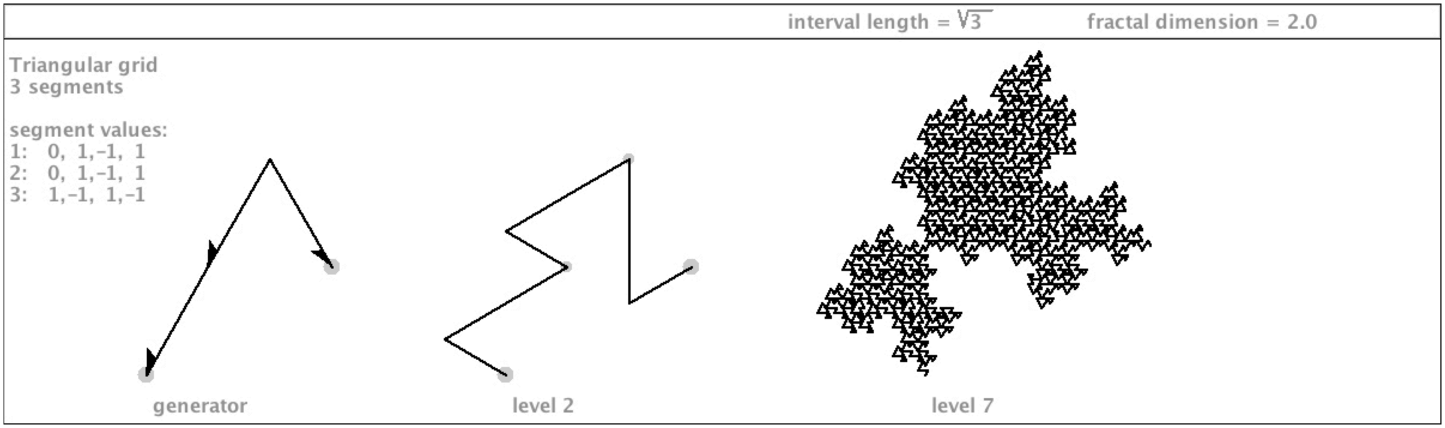


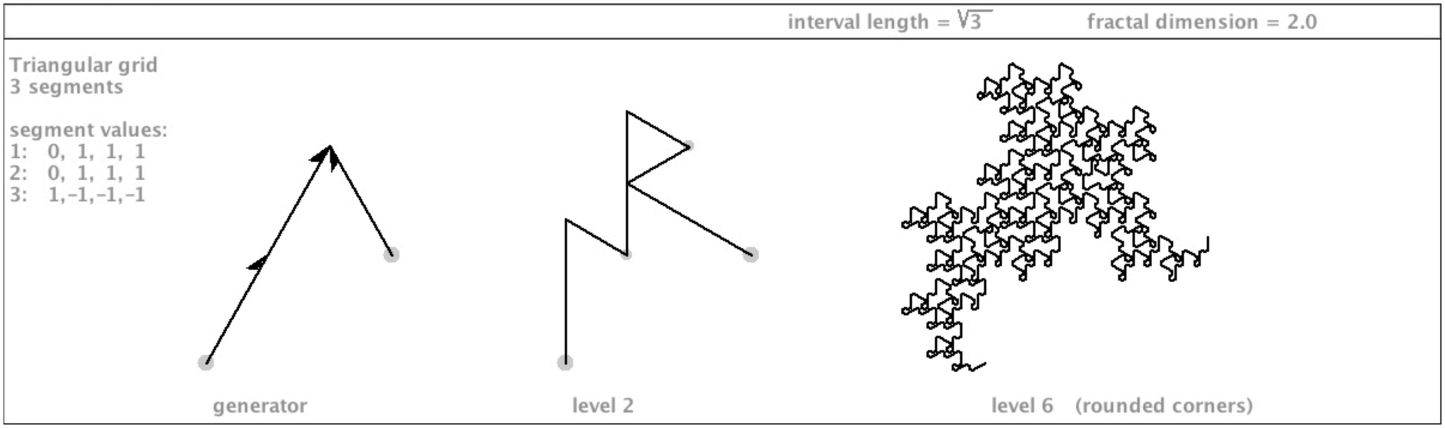
This simple flip changes the resulting fractal curve from being fat in the middle to having a pinched waist. I call it the "inverted Ter-Dragon". Below is yet another variation attained from different flippings of the Ter-Dragon segments. In this case, the first and third segments have their x values flipped. It is hard to predict the outcome of these small changes…and you would probably not have guessed that the result would be a curve that completely fills a rectangle!

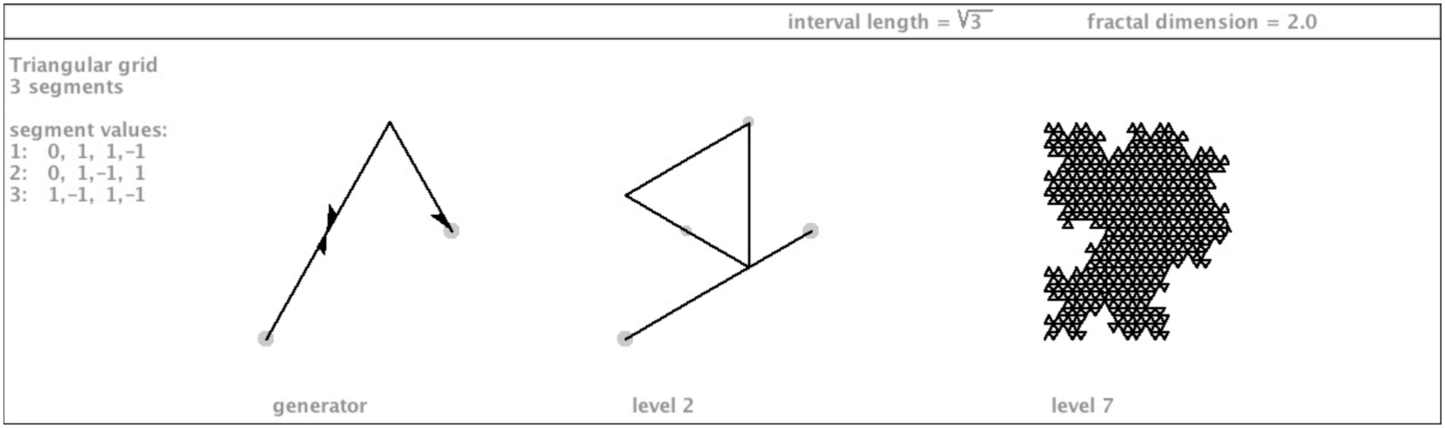


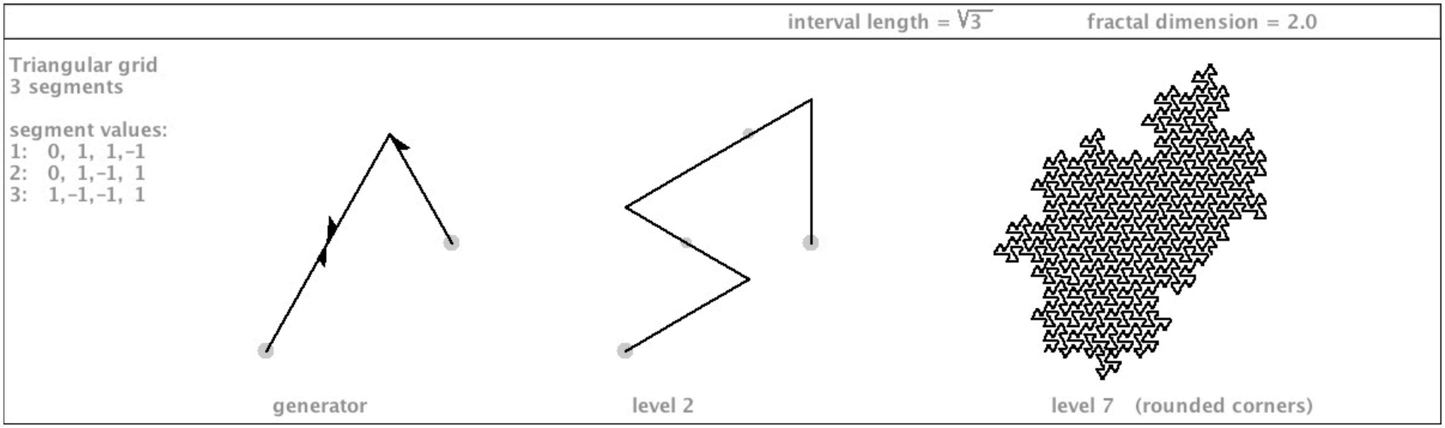
Just ONE flipped number. That's all it takes to transform a craggy-edged butterfly into a box.

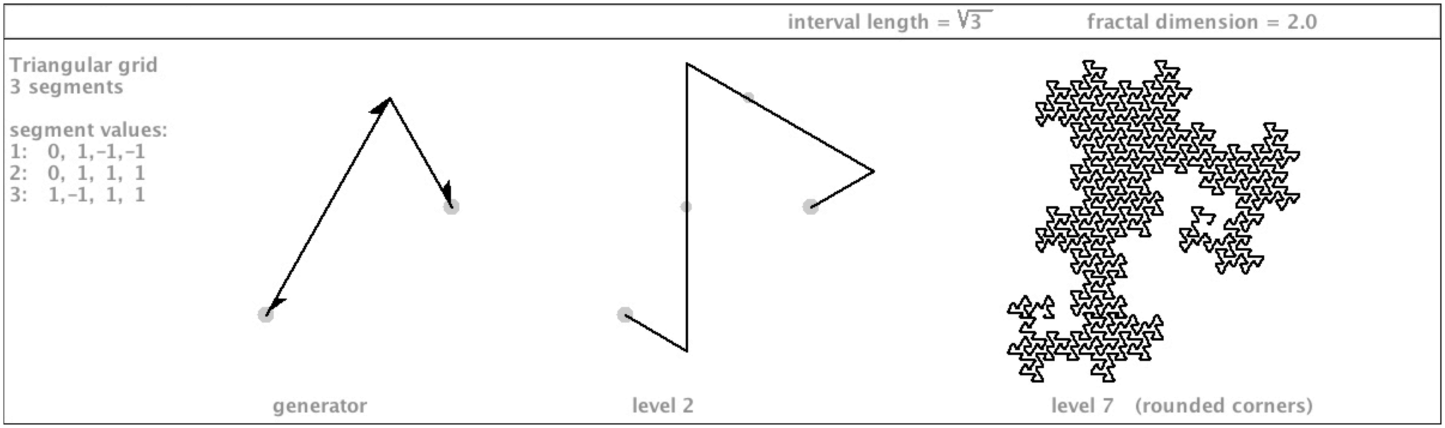
Now let's look at the other kind of path that can connect point A to point B. This one can produce seven unique plane-filling fractal curves. They are shown on the following pages.

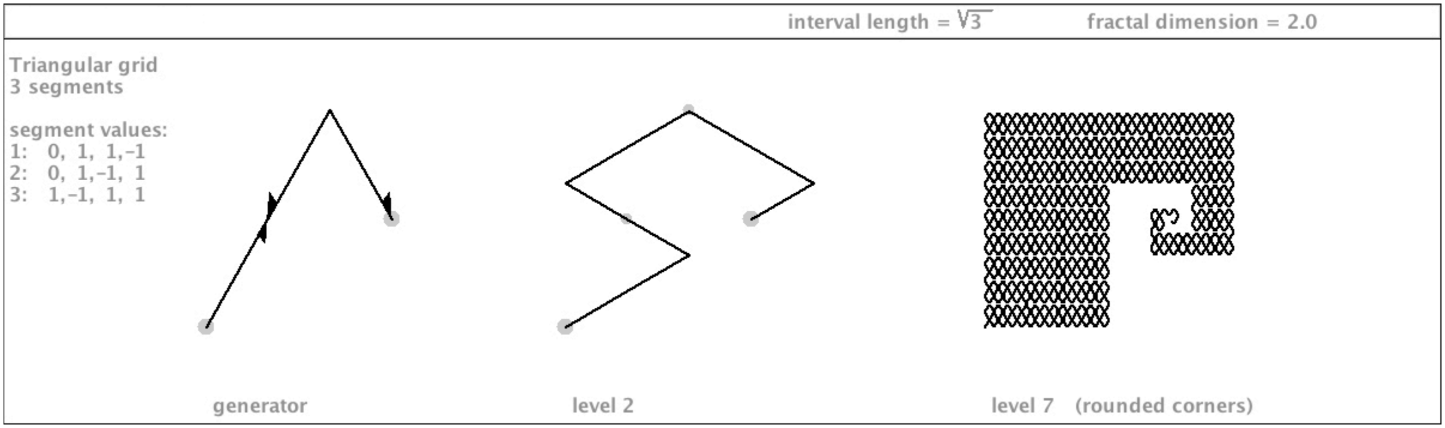

interval length = √3
fractal dimension = 2.0
Triangular grid
3 segments
segment values:
1: 0, 1,−1, 1
2: 0, 1,−1, 1
3: 1,−1, 1,−1
generator
level 2
level 7

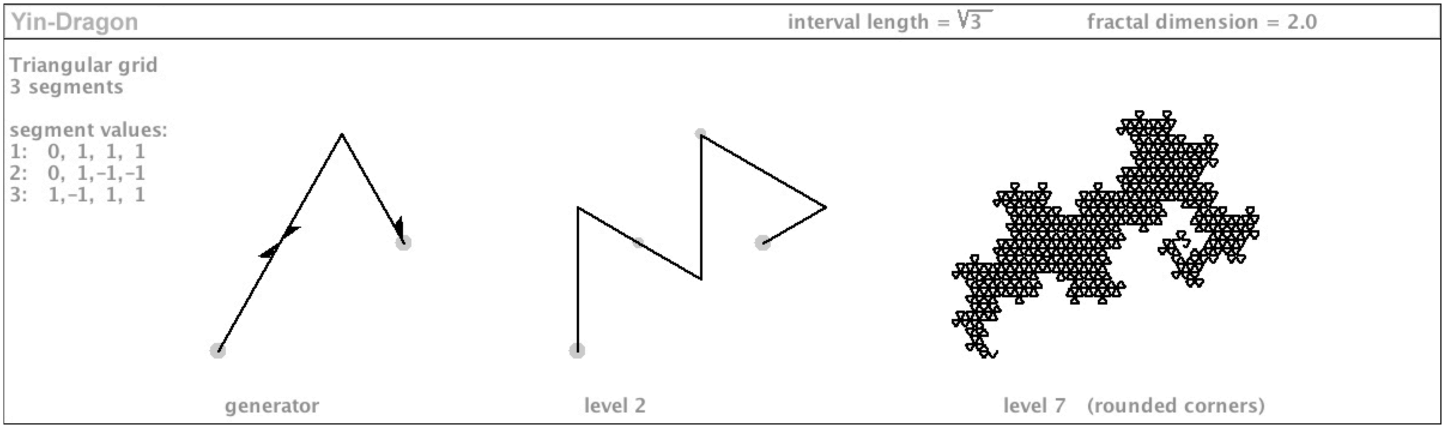

interval length = √3
fractal dimension = 2.0
Triangular grid
3 segments
segment values:
1: 0, 1, 1, 1
2: 0, 1, 1, 1
3: 1,−1,−1,−1
generator
level 2
level 6 (rounded corners)

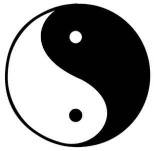

interval length = √3
fractal dimension = 2.0
Triangular grid
3 segments
segment values:
1: 0, 1, 1,−1
2: 0, 1,−1, 1
3: 1,−1, 1,−1
generator
level 2
level 7


interval length = $\sqrt{3}$
fractal dimension = 2.0
Triangular grid
3 segments
segment values:
1: 0, 1, 1,–1
2: 0, 1,–1, 1
3: 1,–1,–1, 1
generator
level 2
level 7 (rounded corners)

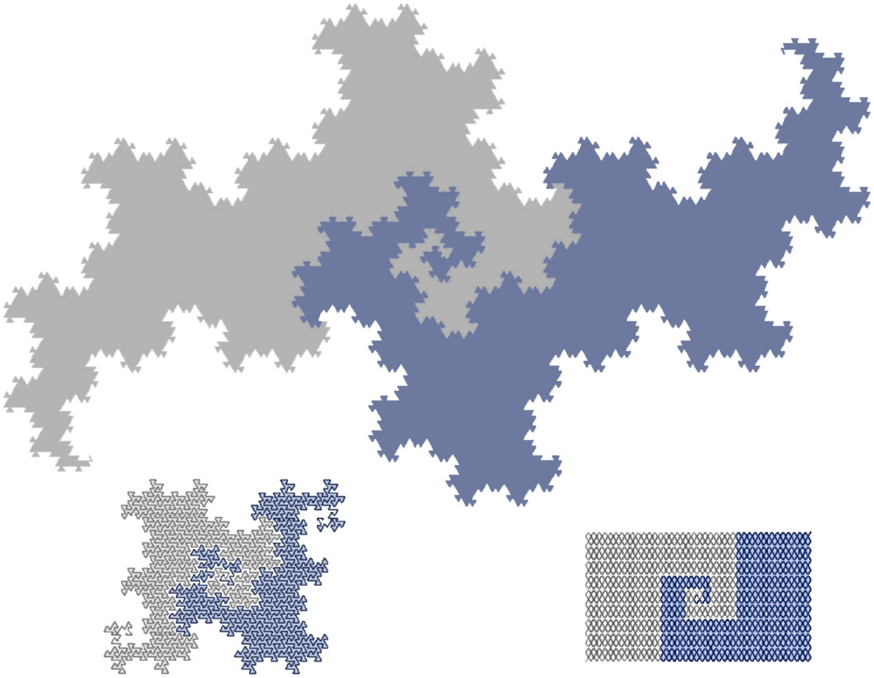

interval length = $\sqrt{3}$
fractal dimension = 2.0
Triangular grid
3 segments
segment values:
1: 0, 1,–1,–1
2: 0, 1, 1, 1
3: 1,–1, 1, 1
generator
level 2
level 7 (rounded corners)


interval length = $\sqrt{3}$
fractal dimension = 2.0
Triangular grid
3 segments
segment values:
1: 0, 1, 1,–1
2: 0, 1,–1, 1
3: 1,–1, 1, 1
generator
level 2
level 7 (rounded corners)

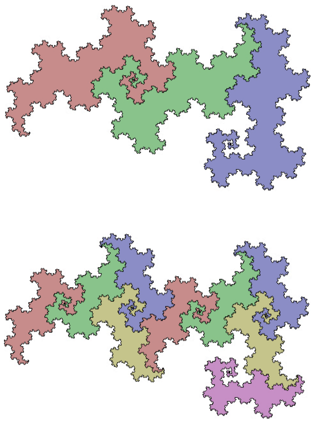




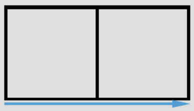
Check it out: the fractal curve above can be combined with another copy of itself that is rotated 180 degrees. When they are joined together they create two intertwined halves of a double-sized Ter-dragon! The more times the teragons are fractalized, the more tightly the handshake in the middle spirals inward. It reminds me a yin-yang symbol. And so, I call it the Yin Dragon. The two previous specimens are also shown at the bottom, as yin-yang pairs.



The Yin Dragon was also discovered by Tom Karzes [11], who called it "Half-TerDragon". He created the following cool pertilings:

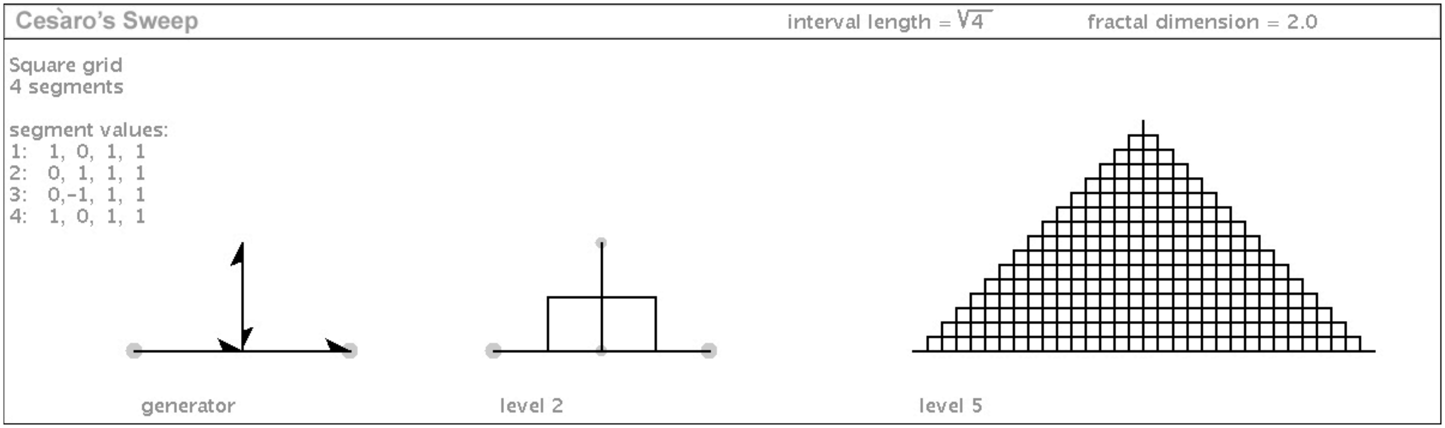


√4

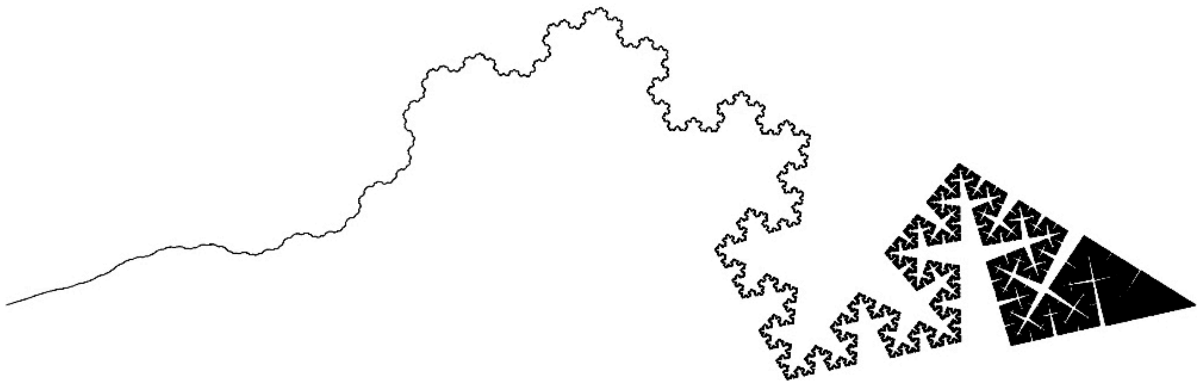


Now we come to the √4 square grid family. Let's return to the Koch Curve, which has four segments. If we morph the Koch generator, progressively sharpening the triangle bump in the middle, causing the first and fourth segments to come closer, the fractal dimension of the resulting curve will approach 2.

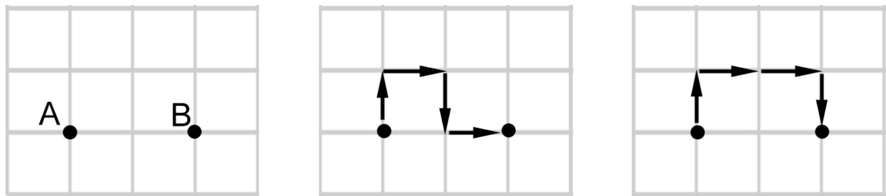
When the transformation is complete, the interval length is 2, and the second and third segments of the generator become a two-sided needle pointing upward. This is a variant of Cesàro's Sweep that I showed you earlier. It is a twice-dense gridfiller: it is self-touching among all segments (except for the segments that lie on the bottom edge).



Wolter Schraa [21] created a nice artistic image showing the transition of a curve from dimension 1 to dimension 2. A Koch Curve-like region emerges just after the middle area, and at the very end, it closes up to form the characteristic folding of Cesàro's Sweep. Here is an altered version of Schraa's image:

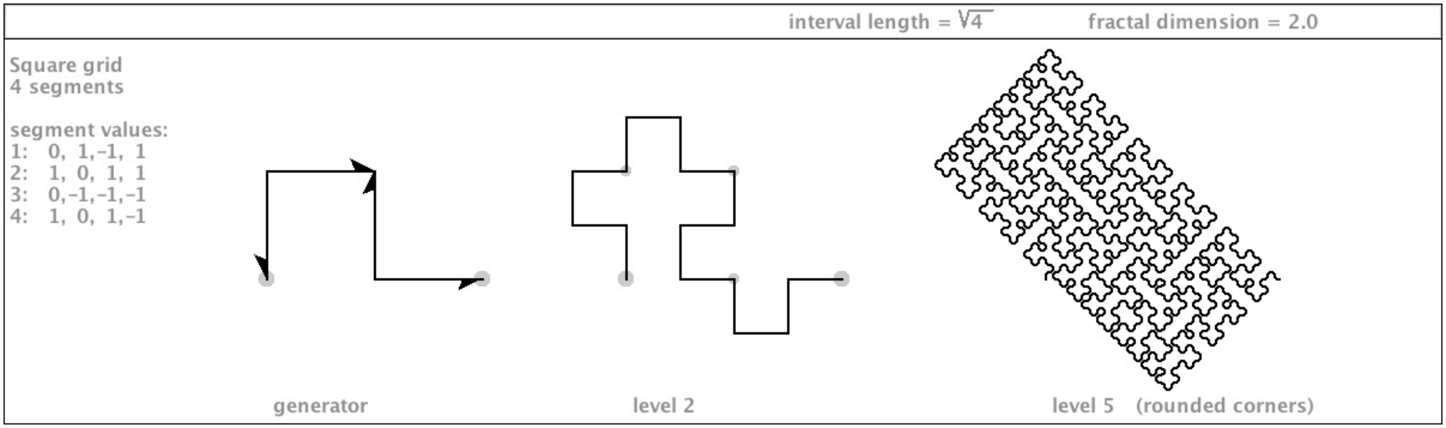


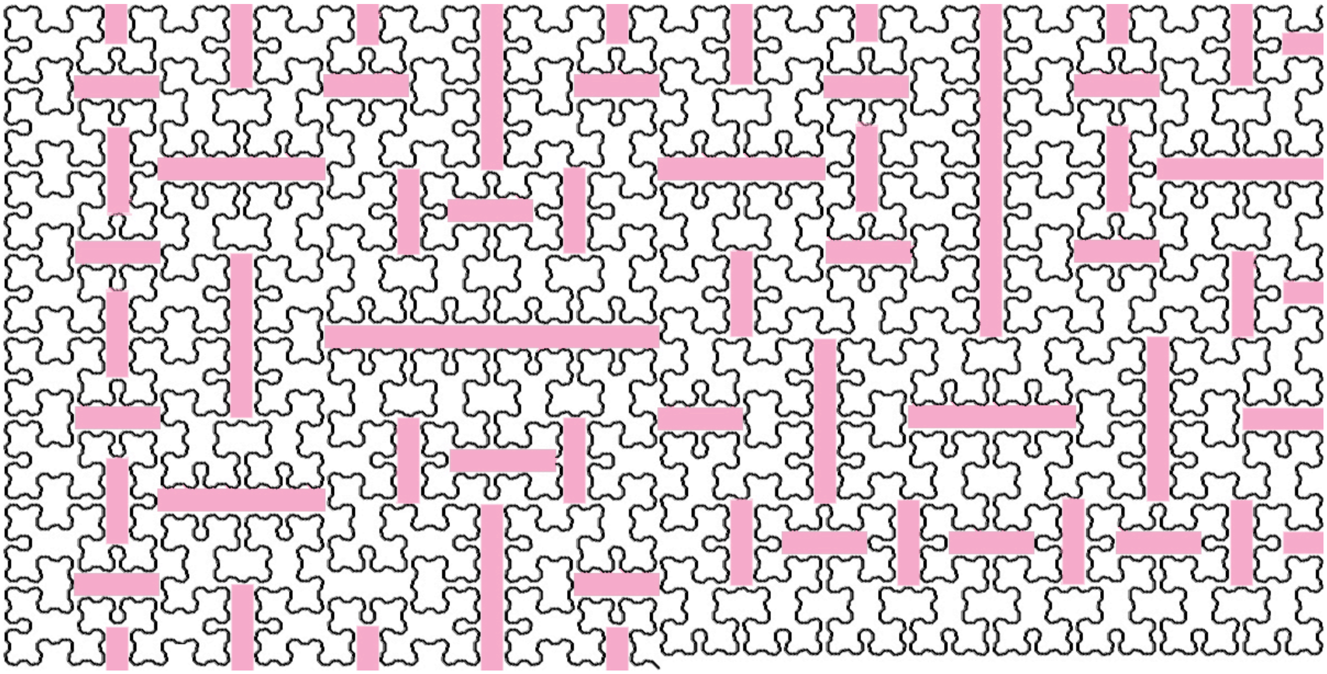
Now, back to a familiar question: in a square grid, how many ways can you draw four connected lines of length 1 from point A to point B, without having them self-contact? (Not counting rotations and mirror-images). The answer is 2:



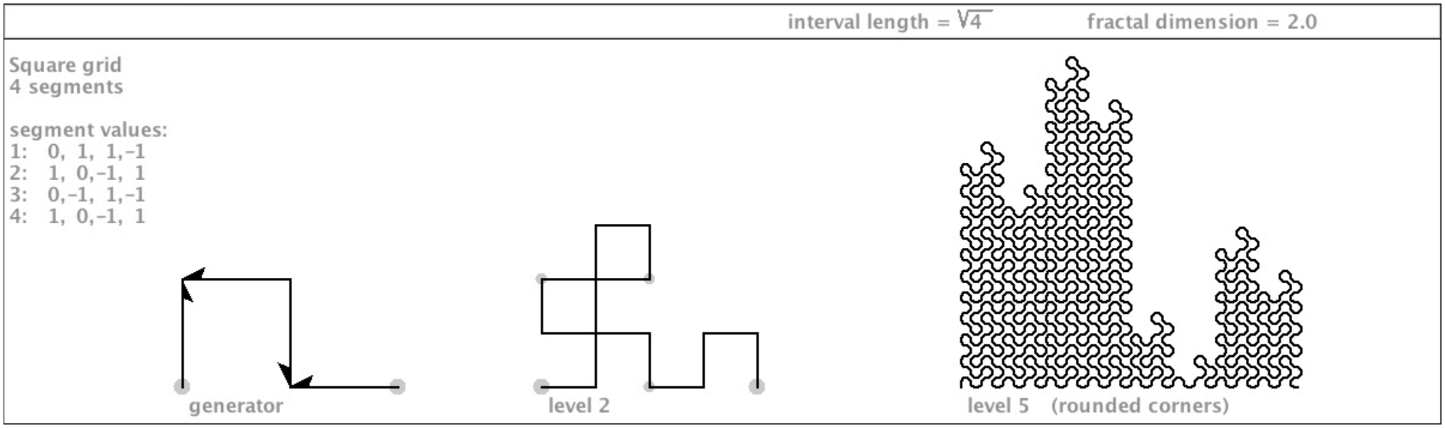
Now, let's start with the first generator above (the square bump with a level segment to its right). How many plane-filling curves can we find using various flippings of this shape? Well, let's rule out a few curves that are already represented by the √2 family: the HH Dragon and the Pólya Sweep. Observe that the level 2 teragons of the √2 family have the same shape (only rotated and flipped). Thus, transformed replicas of the √2 curves can be generated with this shape.

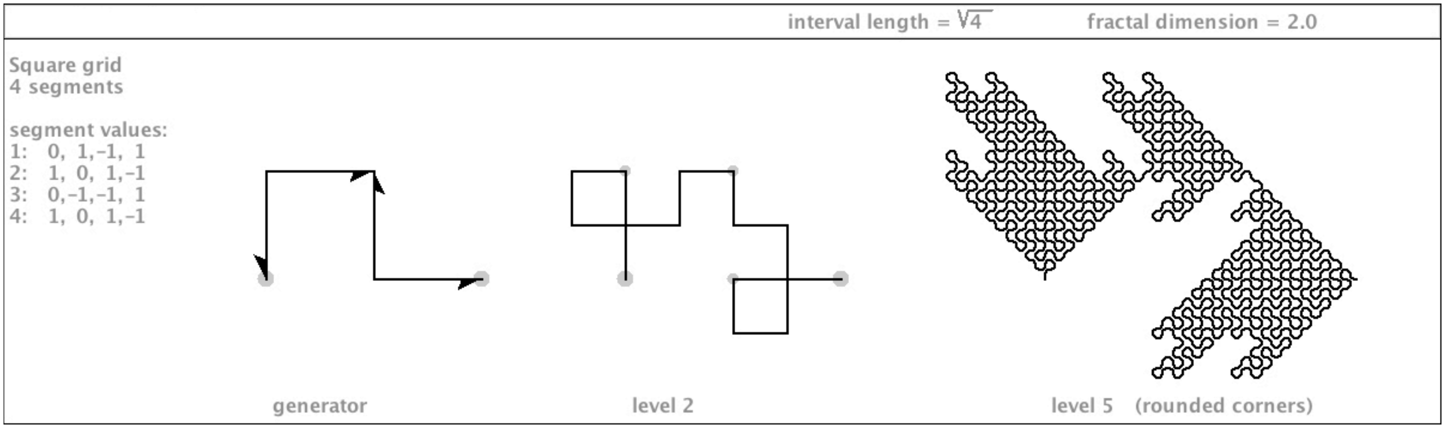
Besides these exceptions, I have discovered three unique plane-filling curves of the √4 square grid family. The first one, shown below, fills a rectangle that is tilted 45 degrees. It is a partially self-touching curve: many parts of the curve are self-touching at their vertices, but the rest of the curve is self-avoiding. It is shown below at level 6, tilted 45 degrees, with colored bars where there are gaps longer then a specific length. Can you describe the pattern of these gaps?



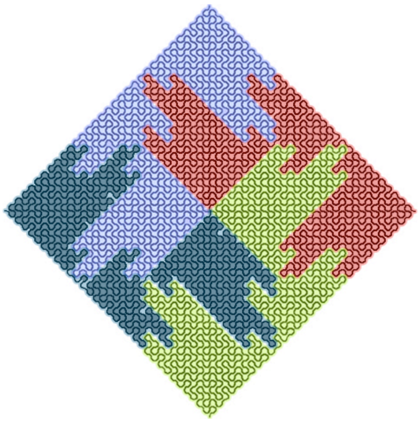


These next two curves are curious indeed. They appear to have variations on a similar motif.

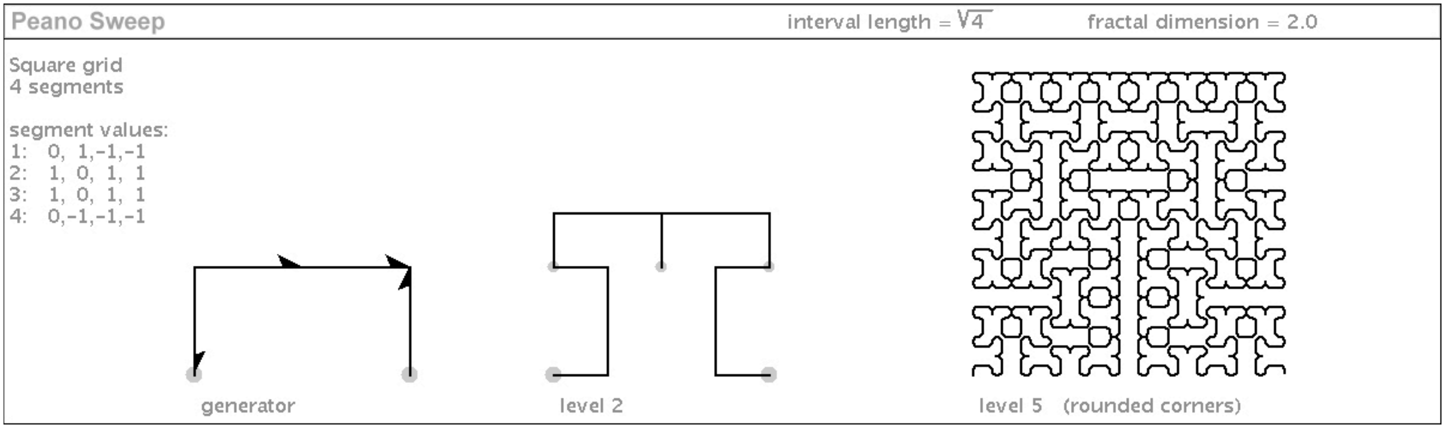




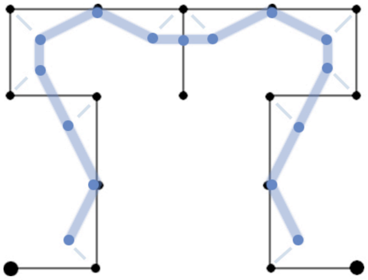
The second specimen is shown at right, pertiled four times, to make a square.

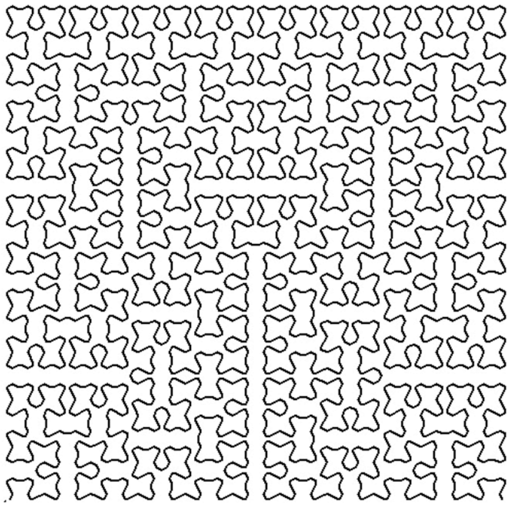


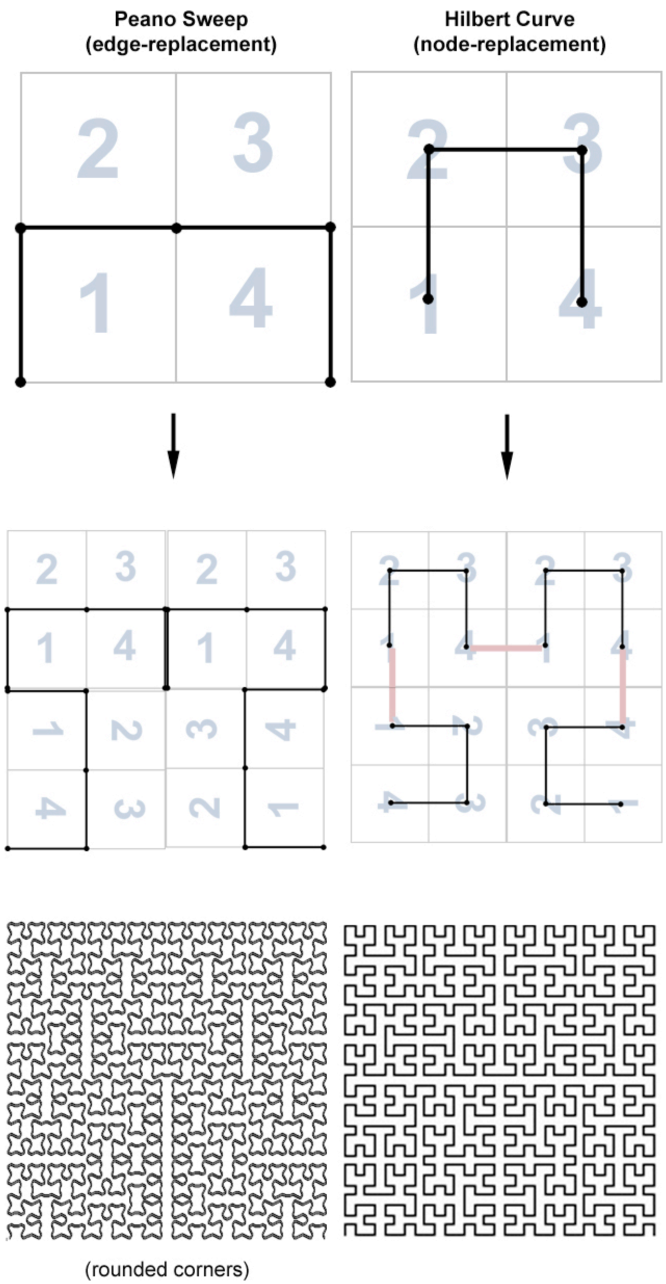
Now I'll show you the other curve that can be made using the other path from A to B that I showed. Mandelbrot included it in his book, and attributed it to Peano, calling it the "Peano Sweep". It is self-touching along some proportion of its edges. This is apparent at level 2.



For curves that are self-touching on edges, I sometimes use a different technique than rounded corners. In this case, I use a simple low-pass smoothing filter. Basically, after the points of the curve have been calculated, I adjust the position of each point to equal the average of the positions of itself plus its two neighbors. This has the effect of separating the edge-adjacencies, allowing space around the curve to breathe. Did you notice a similarity in the inner pattern to an earlier specimen of this family?







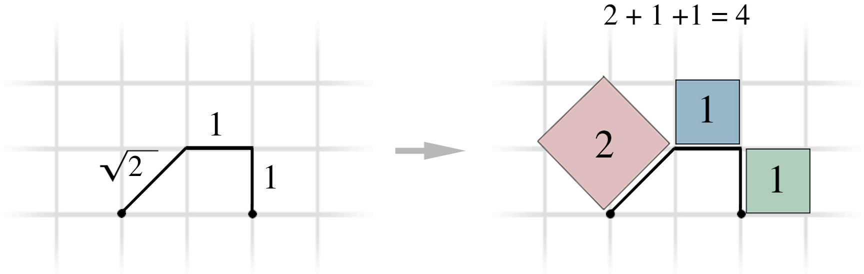
After applying this technique to make the curve breathe, I noticed a similarity to higher-level fractalizations of the Hilbert Curve, which we met earlier. And indeed, there is a direct correlation. The Peano Sweep is basically a *Koch-constructed* variation of the Hilbert curve, using edge-replacement instead of node-replacement. Notice that in both cases there are four tiling squares.

In the second teragon, we see that the scaled-down copies of the generators that are applied to the two bottom squares are pointing inwards. The copies of the generators applied to the top two squares remain pointing upward, in their original orientations. As I pointed out before, node-replacement requires extra connective lines, shown here in pink.

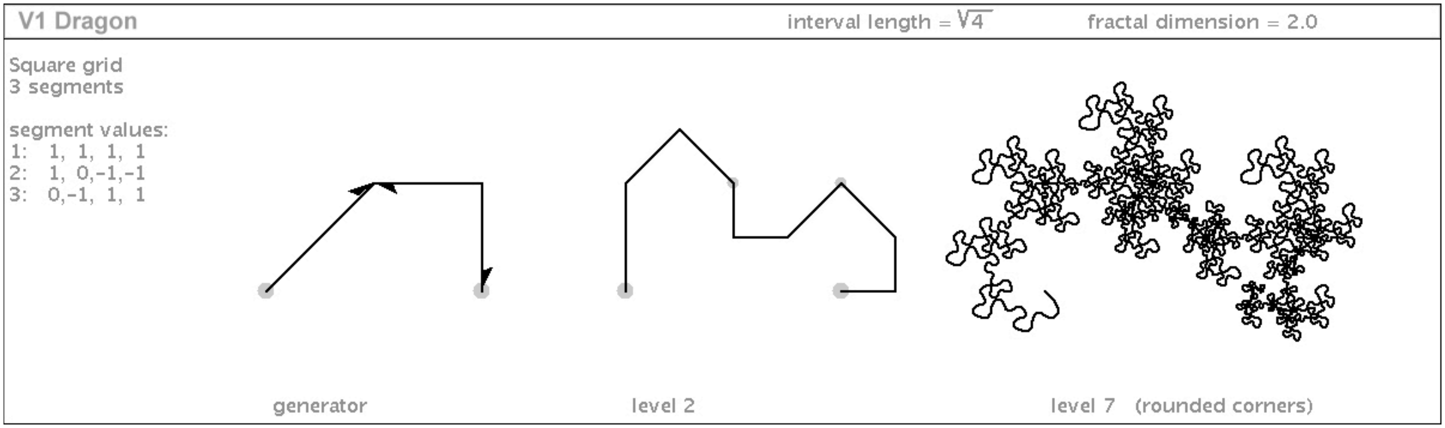
Also, notice that after several levels of fractalization, these curves start to look more similar (when using the smoothing filter technique on the Peano Sweep). Check out the main artery extending upward from the middle of the bottom, which is more noticeable on the Peano Sweep. There are also several secondary arteries – all of which correspond to the cascade of transformations used to generate the curve.

A New *Slant* on Fractal Dimension

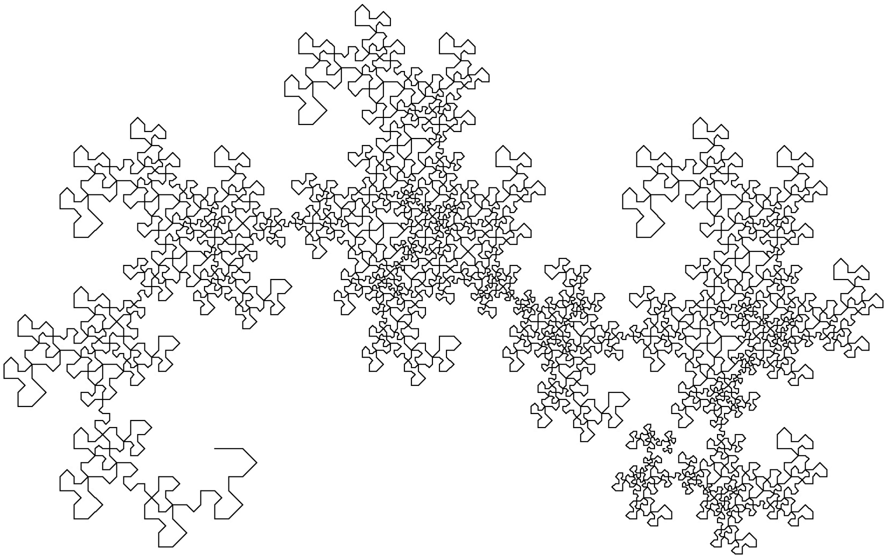
Now it is time to explain a new aspect of fractal dimension, which wasn't necessary until now. Consider the illustration below. The shape at left has three segments, but the *slanted* one is longer than the others. Its length is √2, while the other segments have length 1. Now here's a trick: if each of these lengths are squared, and then summed, the result is 4.



Might this generator create a curve that qualifies as a member of the √4 family? Can a curve of the √4 family with only 3 segments fractalize to a plane-filling curve? Well, let's revisit the equation for fractal dimension: **log N / log L.** Now, instead of N representing the number of segments in the fractal generator, let's re-define N as: "the summed squares of all the segment lengths". Up until now, all the generators so far have had segment lengths of 1, and since √1 = 1, we could just refer to the number of segments. But now, we will change the definition of N to accommodate segments lengths greater than 1. And behold: the generator I have just showed you, given just the right flippings, results in a fractal curve that looks like the HH Dragon (but not quite! – look more closely). I call it *V1 Dragon*.



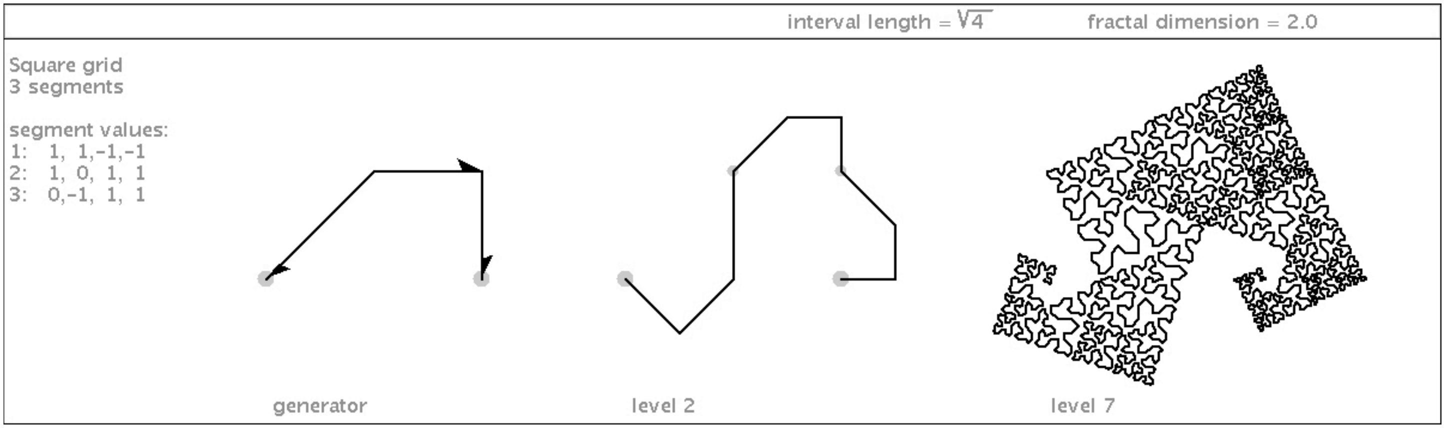
Notice that some of the blobs are bigger than others. That is because of the difference in lengths among the segments, which cascade into many different sizes. The longest segments appear to be at the left-bottom (at the start of the curve). This is related to the fact that the first segment in the generator is the longest.



But is it Plane-filling?

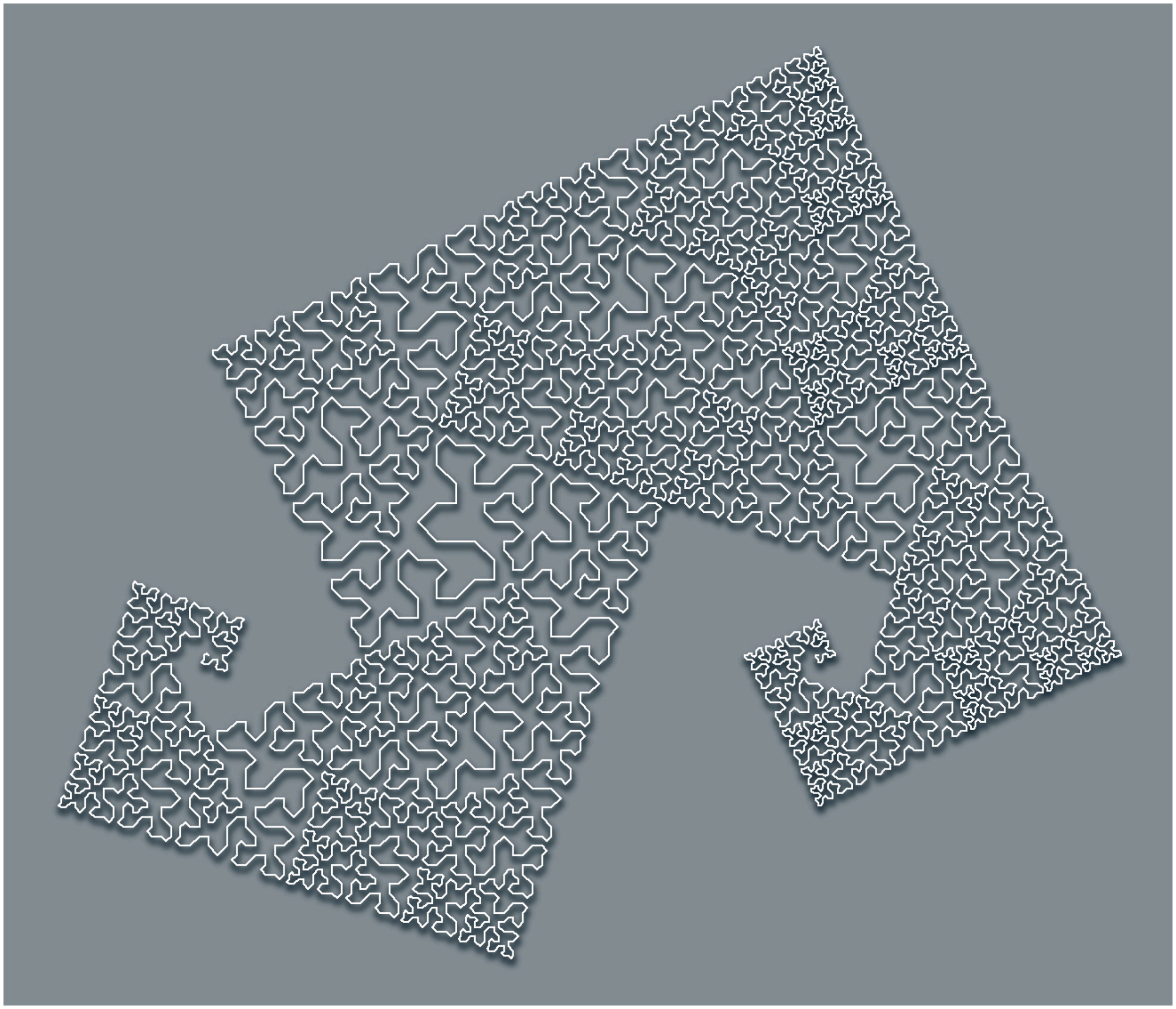
You might not think this a plane-filling curve, because of all these conspicuous blobs of different sizes. But remember that this curve (and in fact *every* curve in this book) has a limited fractal level. If we were to fractalize this curve to infinity, all of the curls would accumulate and close up to completely fill the shape. BUT…even at infinity, would the density still not be uniform throughout the shape? I shall leave this as an open question for you to ponder.

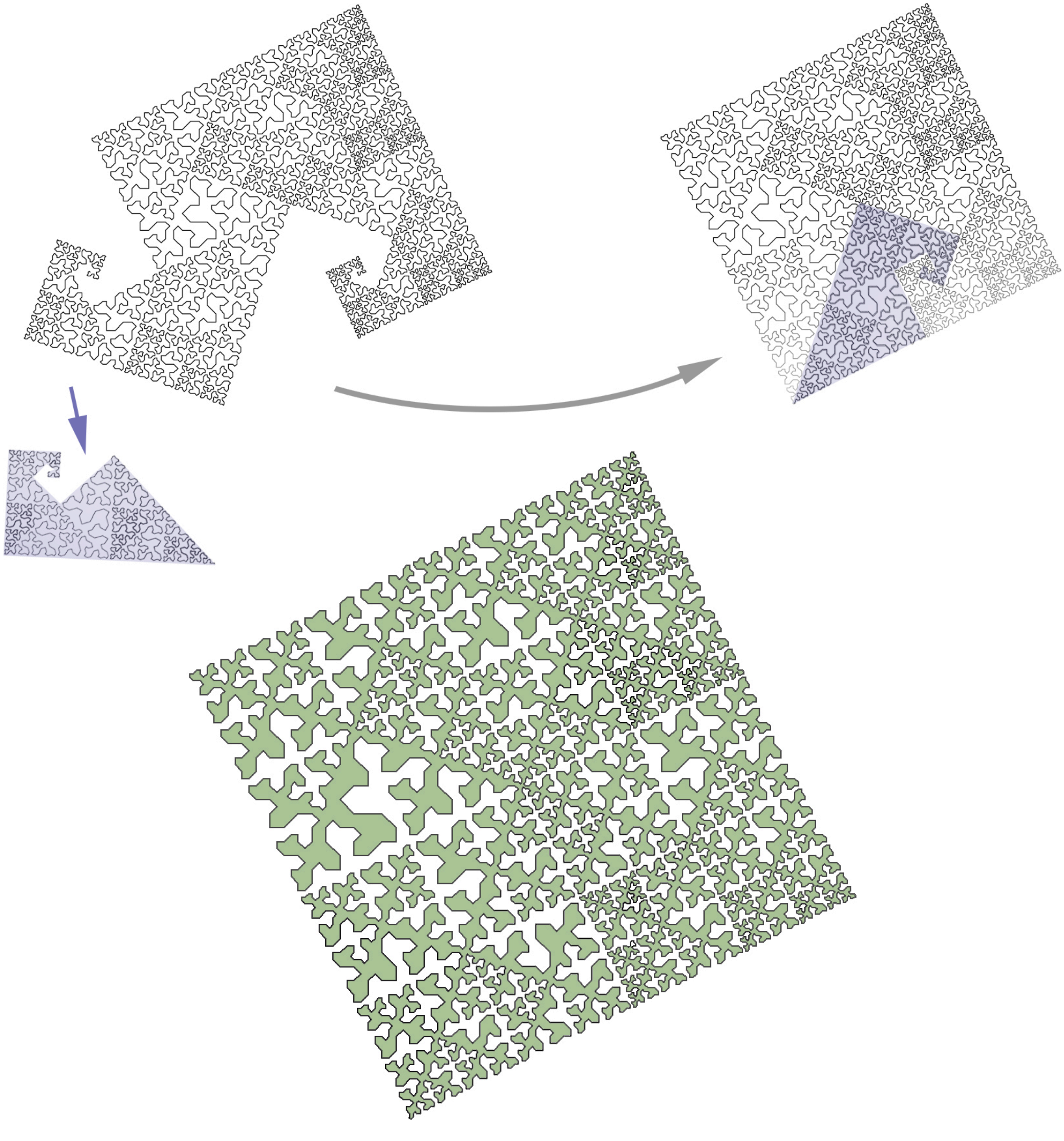
Now let's try a different flipping of this shape. Lo and behold, the resulting fractal curve looks quite different indeed.



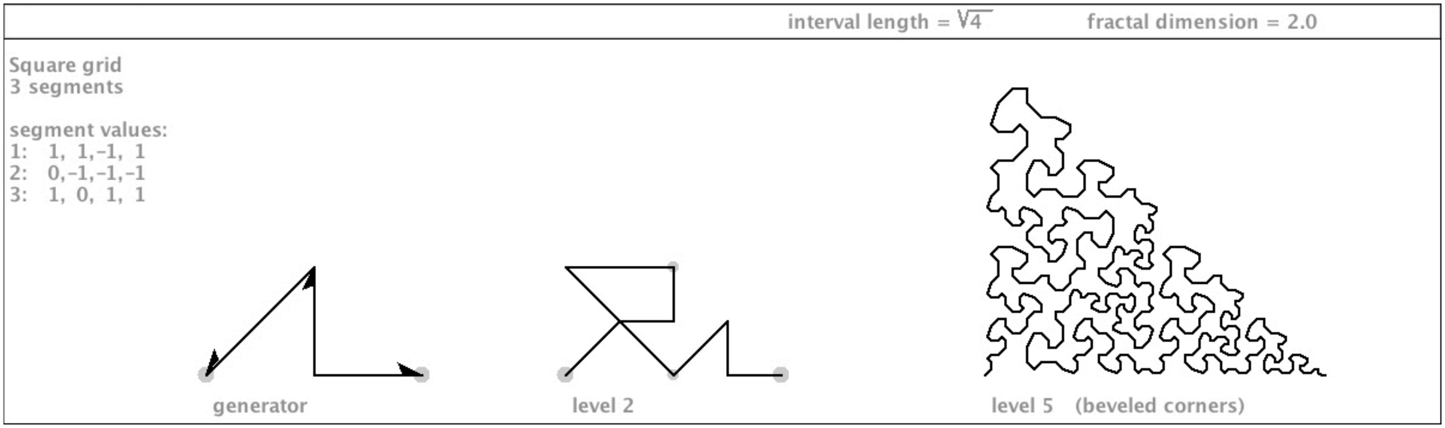
This fractal curve is special. It is not too often that I find fractal curves that are self-avoiders. Well, this one is! Let's see it enlarged a bit, and with some cool coloring added. By the way, this curve appears to also have been discovered by artist Víctor Carbajo [3].

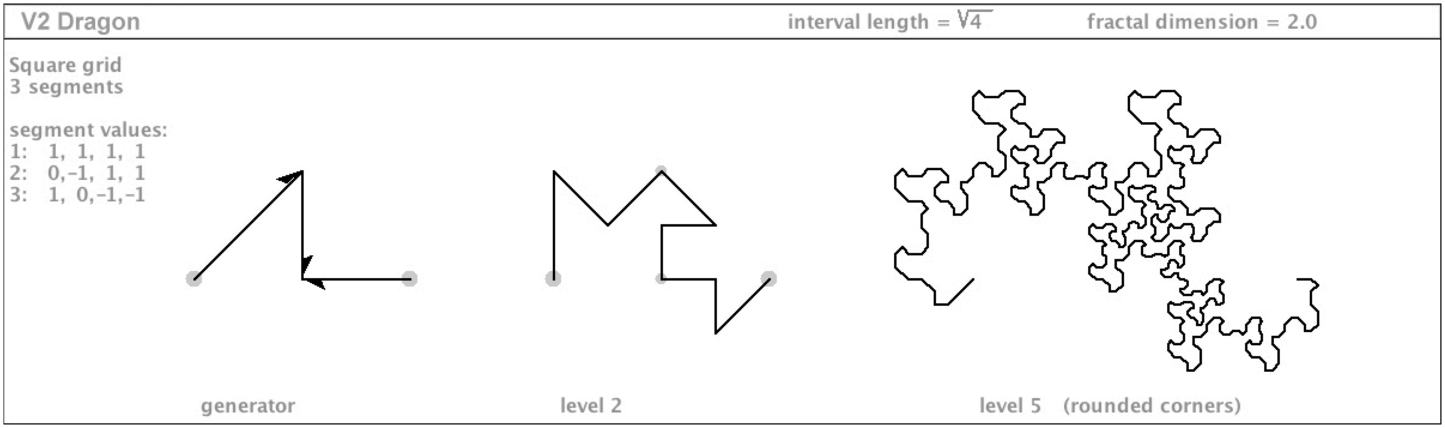
And here's an interesting fact: if you detach the section at the lower-left, and rotate it 90 degrees, pivoting about the bottom corner, it fits snuggly into the remaining hole. Not only that, but it turns the curve into a closed loop. On the next page, I show this process, and then I show what it looks like with the interior of this closed loop filled with a solid color.





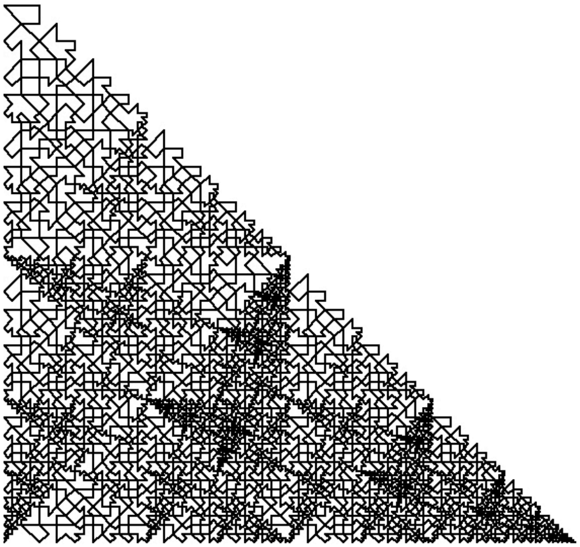
Below are two other generators that have a segment length of √2.

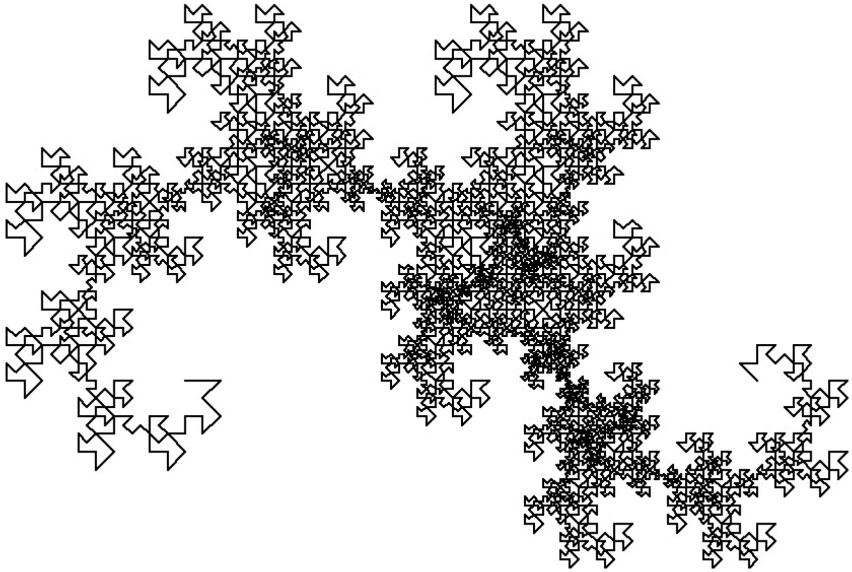




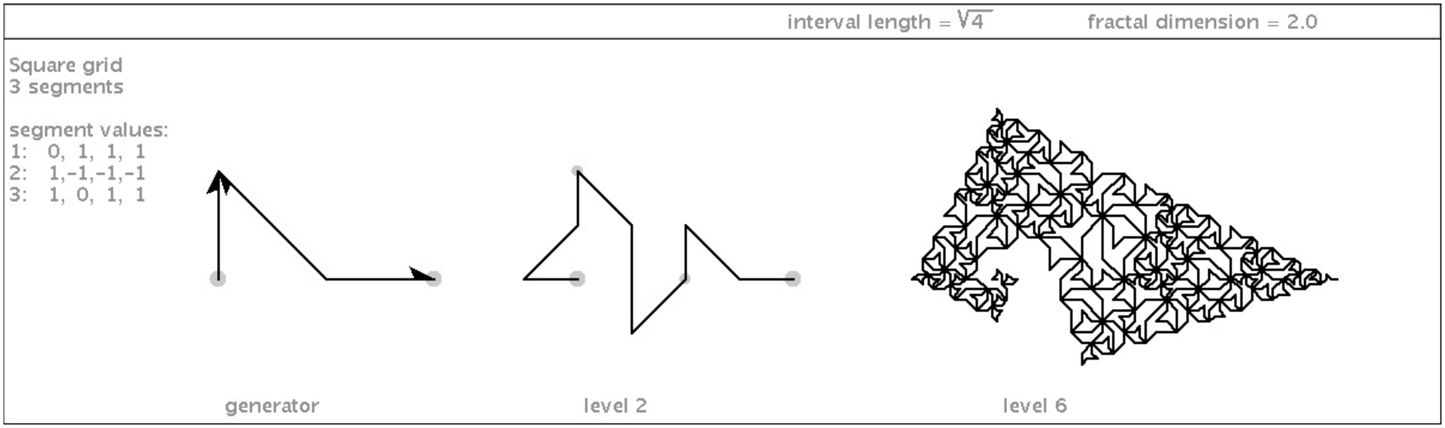
Look familiar? Given a specific generator shape, one set of flippings results in a right triangle while the other results in a dragon (I call this one the *V2 Dragon*).

Both of these fractal curves are partial gridfillers: Some of their vertices touch and some don't. This makes for some interesting internal patterning, as shown on the next page.

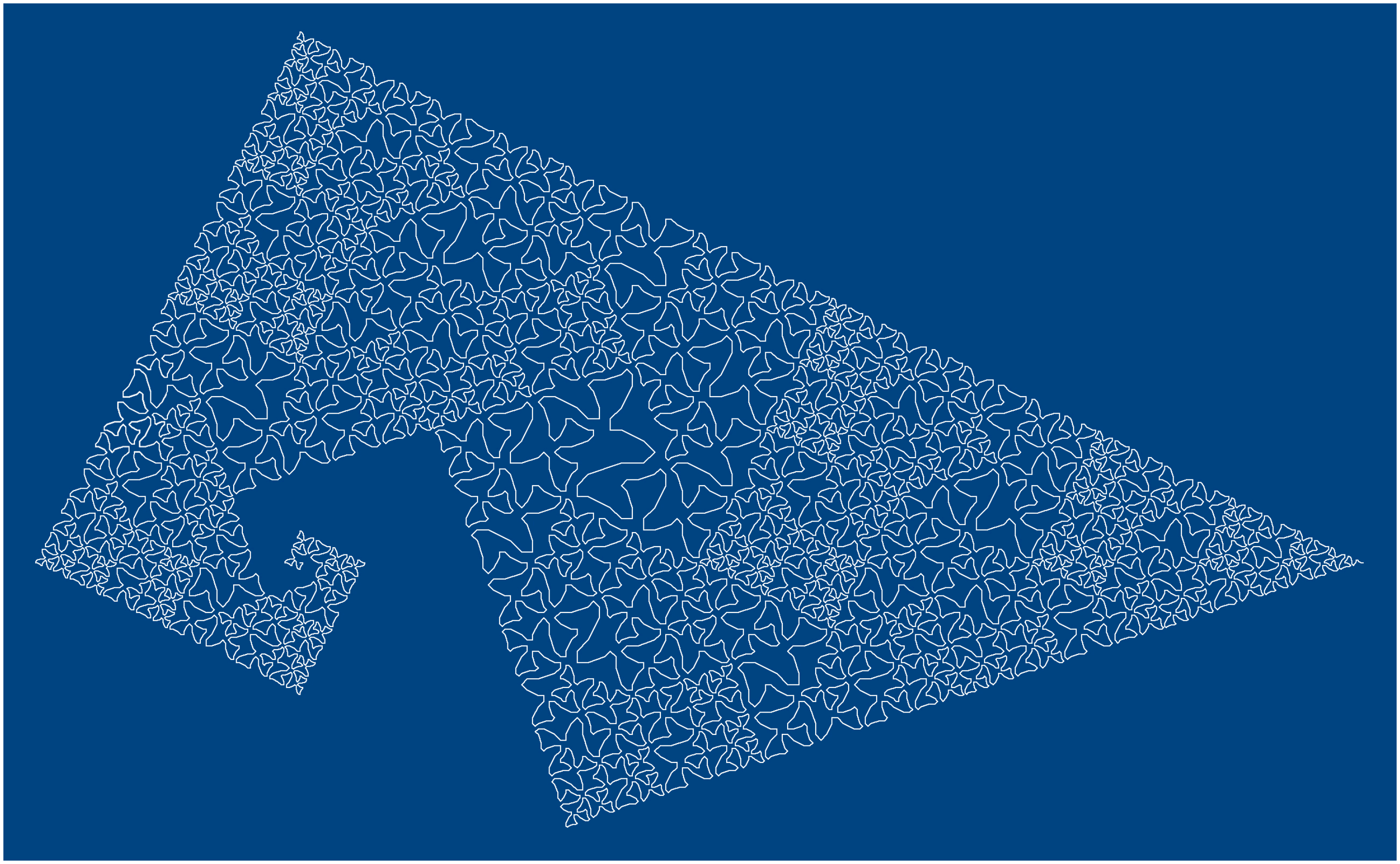




There are two more generators of this family that have a segment length of √2. Here’s one of them:

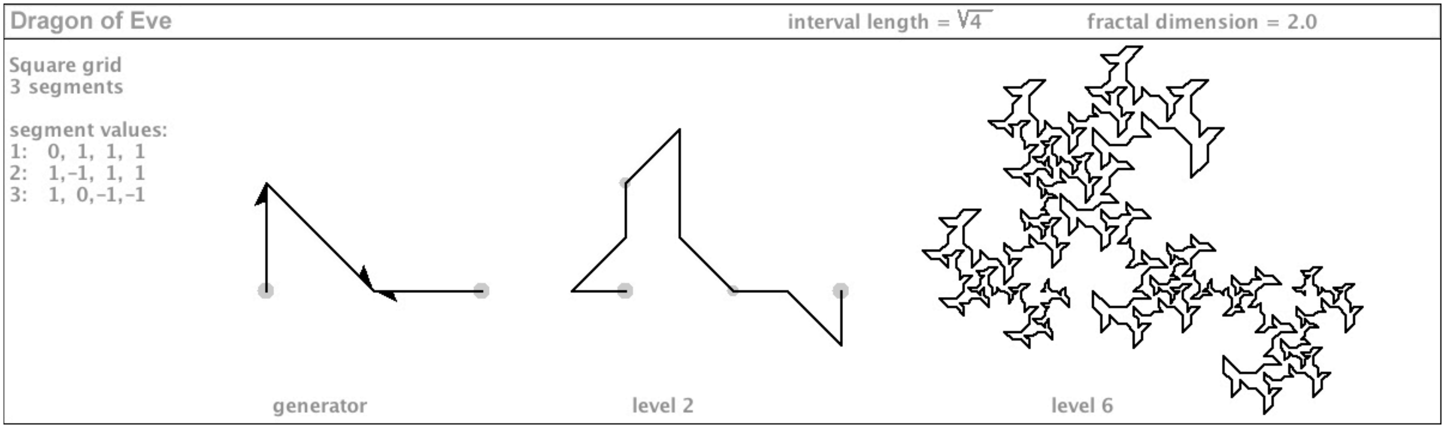


On the next page it is shown enlarged at a higher level, and with rounded corners, so you can appreciate its meandering path.

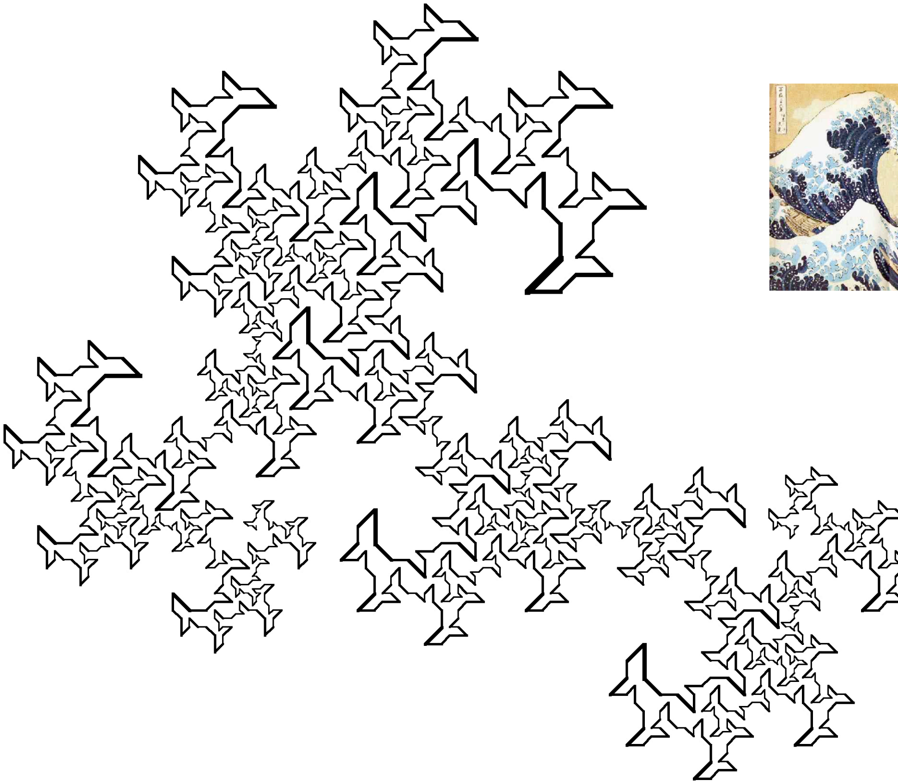


Finally, I want to show you a member of the √4 square family, which I am especially proud of. I call it the “Dragon of Eve”. It is named after Eve Peters, who was my mother’s Art teacher in High School, and whose house I stayed at in my first semester in graduate school. I discovered it while living in her house.

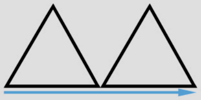
The Dragon of Eve is a self-avoiding fractal curve. Here it is:



Here it is enlarged at a higher level. This drawing uses a technique in which line thickness is proportional to line length. This curve reminds me a bit of the Great Wave of Kanagawa, by the Japanese artist Hokusai.

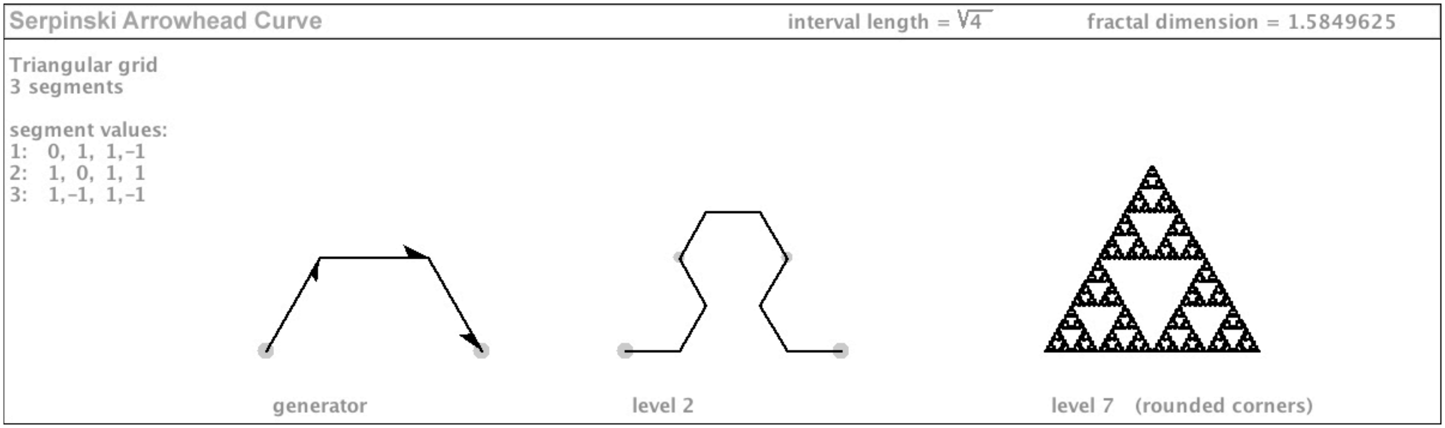


$\sqrt{4}$

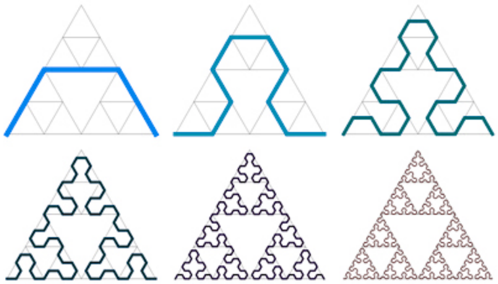


Since √4 = 2, and since square grids and triangle grids share common grid points along an axis (the horizontal "floor" of the two grids), you may suspect that there exists a family of √4 plane-filling curves that live in the triangle grid. Indeed I have discovered what I believe to be all of them.

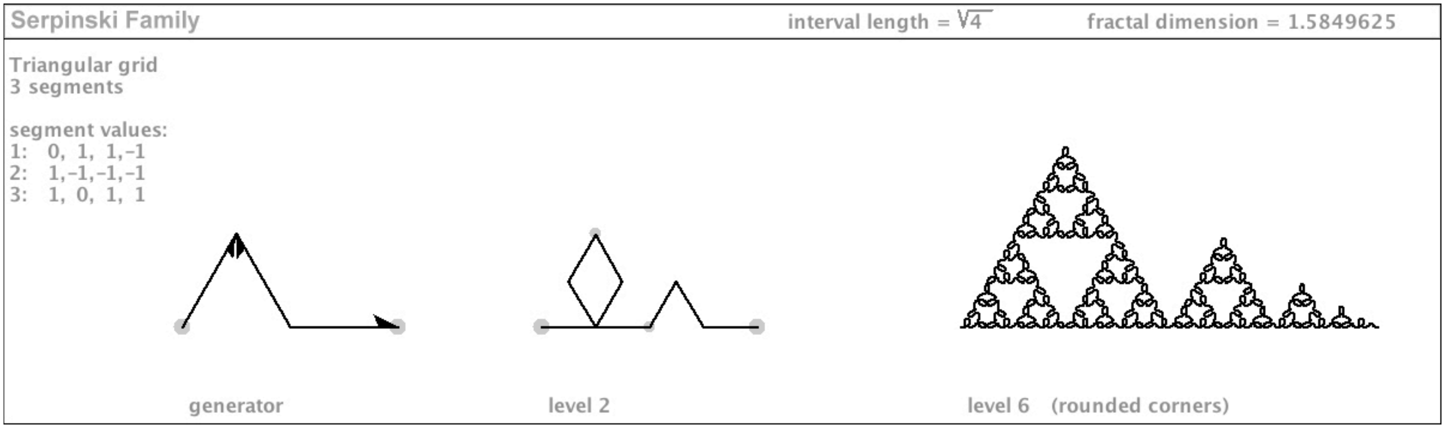
But before I show you these plane-filling curves, I would first like to show you a member of the √4 triangle grid family that is not plane-filling: its fractal dimension is ~1.5849625, and it generates the famous Sierpinski Arrowhead Curve. It is identical to the *Sierpinski Triangle* (a solid triangle with its center triangle cut out, and then with the center triangles cut out of the remaining three, and so on)...



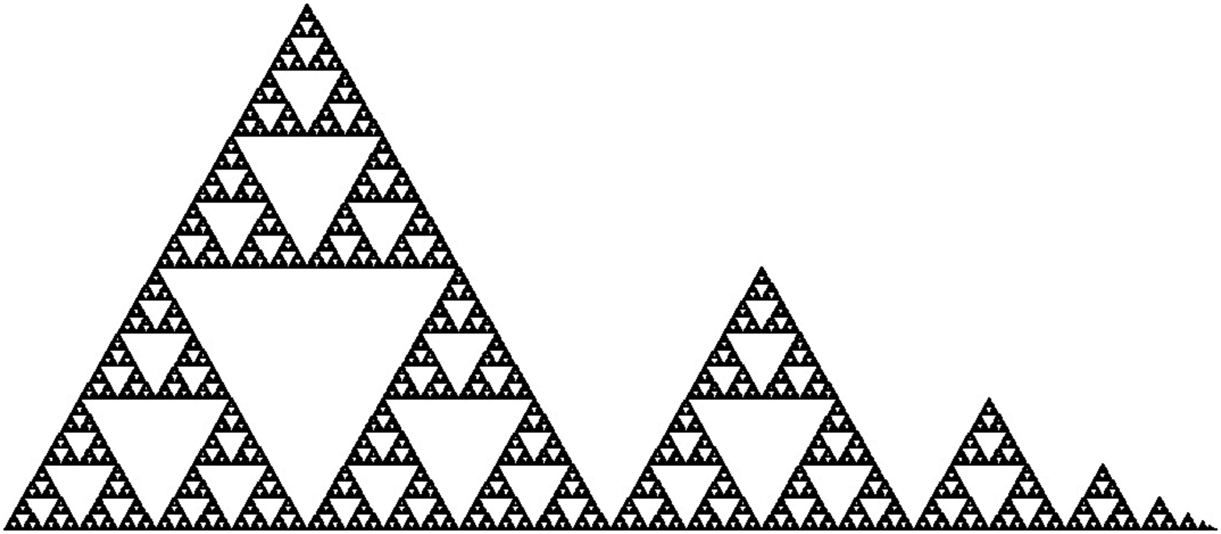
When you fractalize the Sierpinski Arrowhead Curve, it converges toward the Sierpinski triangle. At each stage, it accumulates bays and peninsulas, which approach each other, getting closer and closer….but never touching. So in fact this is a self-avoiding curve.



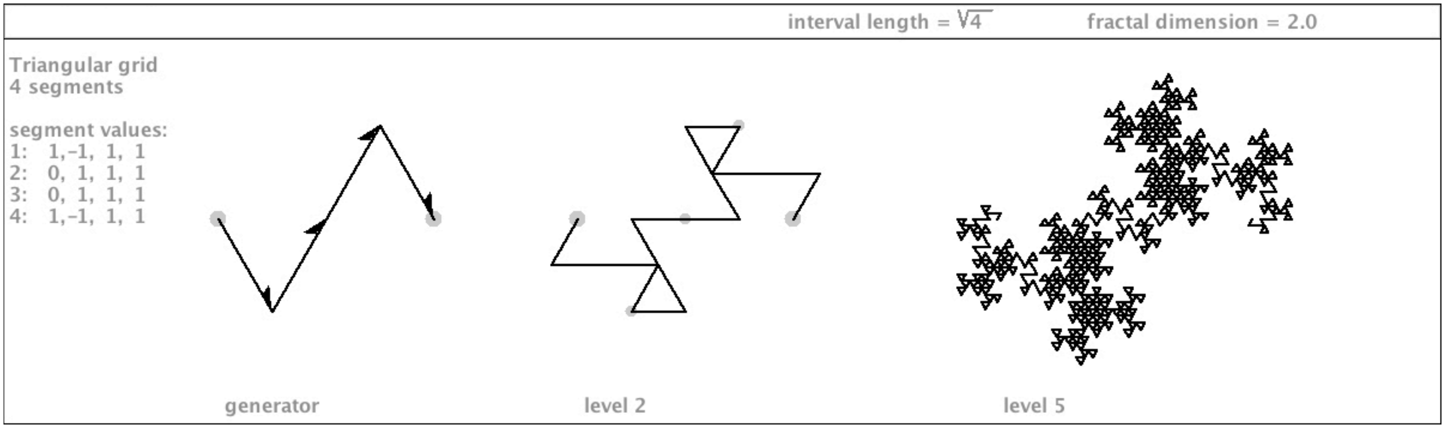
I was delighted when I discovered a curve that actually creates a *family of Sierpinski curves*! In the diagram below, I show this curve fractalized to level 6. Daddy Sierpinski sits proudly to the left, with Mommy Sierpinski to his right. To her right is daughter Sierpinski, followed by little brother Sierpinski, and then baby Sierpinski, and finally, the family pet: Turtle Sierpinski.

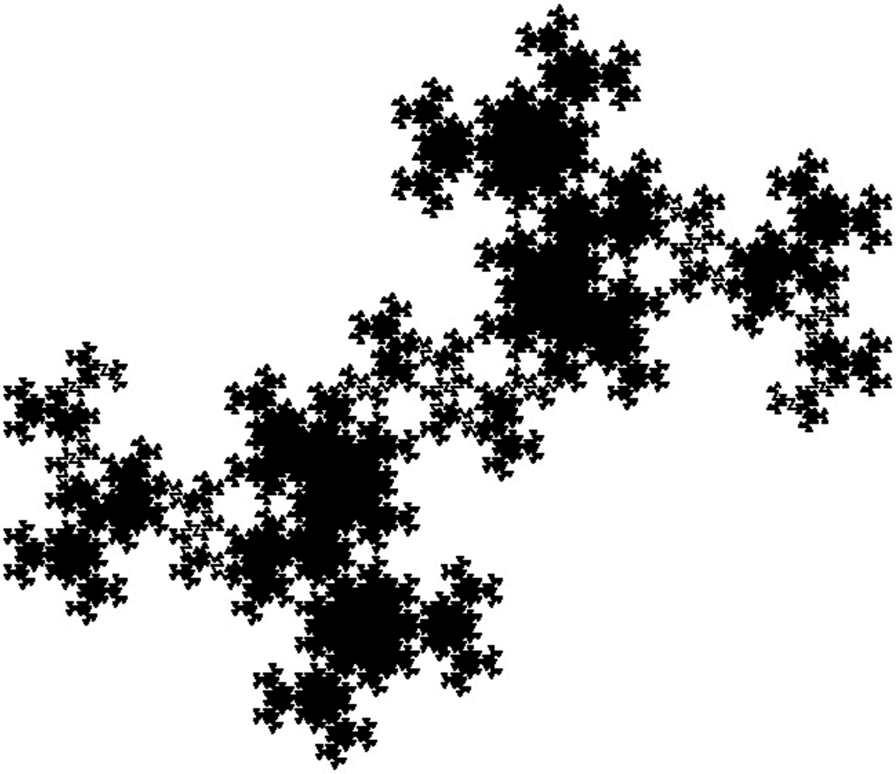


Here is a higher-level rendering of the Sierpinski family fractal curve, with some extra members of the family to the right of the pet turtle (I leave it to you to imagine who are the tiniest members of this family).

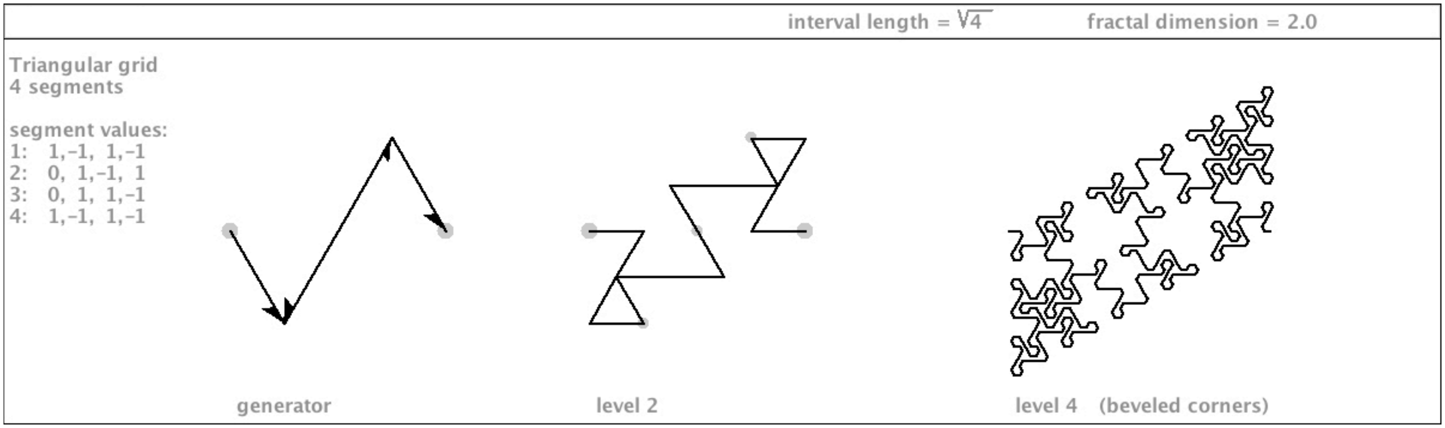


Now it's time to look at the plane-filling members of the √4 triangle grid family. I will start with one that requires no flippings in its segments. Here it is. It happens to be a palindrome:

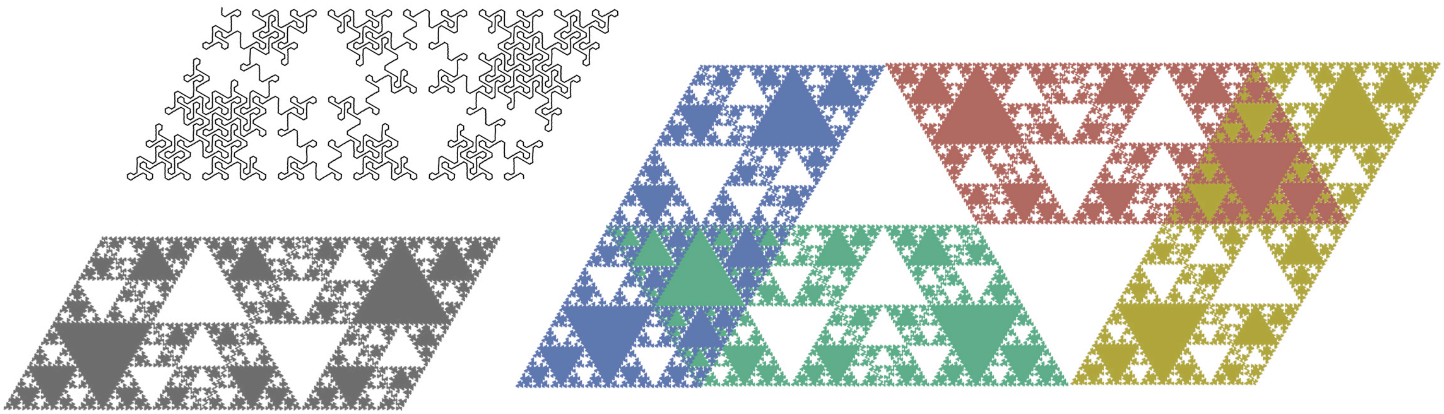




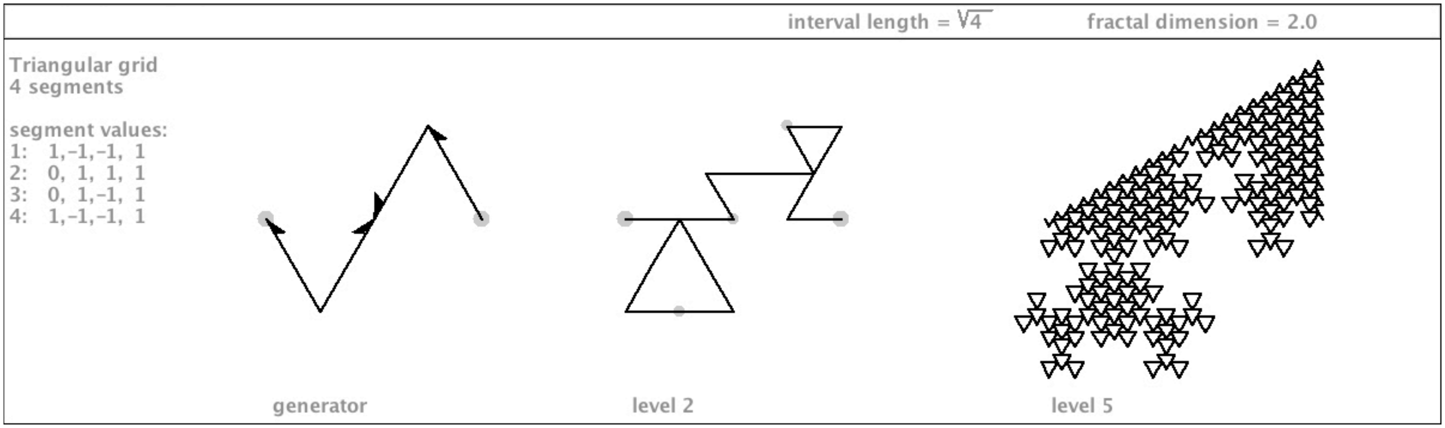
This fractal curve is a gridfiller, but it has a wild boundary, which has a high fractal dimension of its own. Not only that: the boundary touches itself. This fractal makes the notion of "plane-filling" very fuzzy (so to speak) because the region of the plane that it fills is scattered haphazardly. All the filled-in areas of fully fractalized curves that we have seen before this one had boundaries that were either straight lines or else they were fractal curves of their own…but never self-touching. Now get ready; here is that same generator with some different flippings. Its boundary is so amazingly self-touching, you might call it "self-enveloping" (but it is NOT self-crossing, as revealed by rendering it with rounded corners).



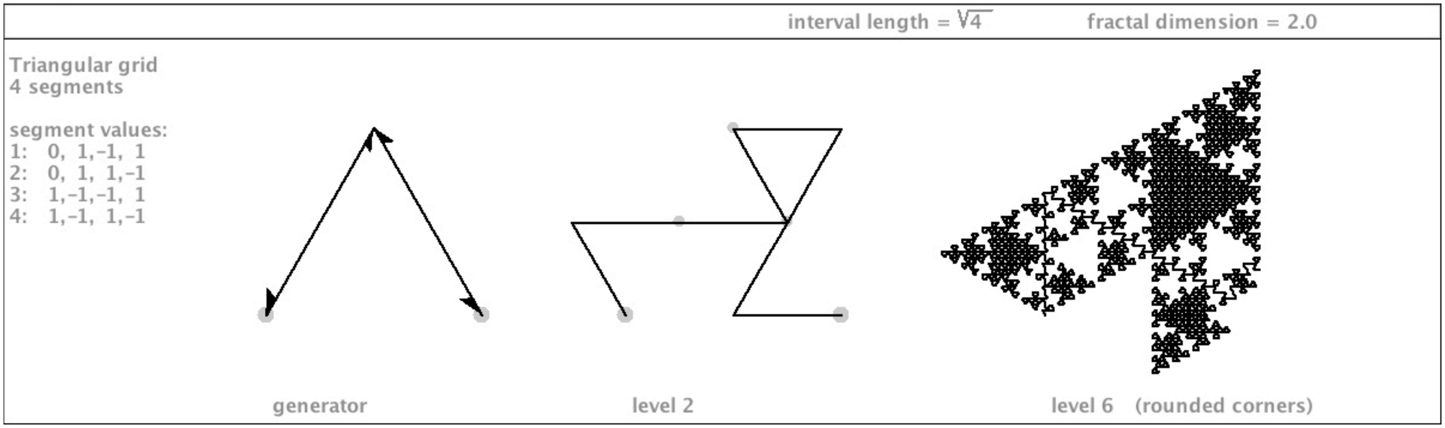
When highly fractalized, this curve becomes a parallelogram filled with a cacophony of triangles.



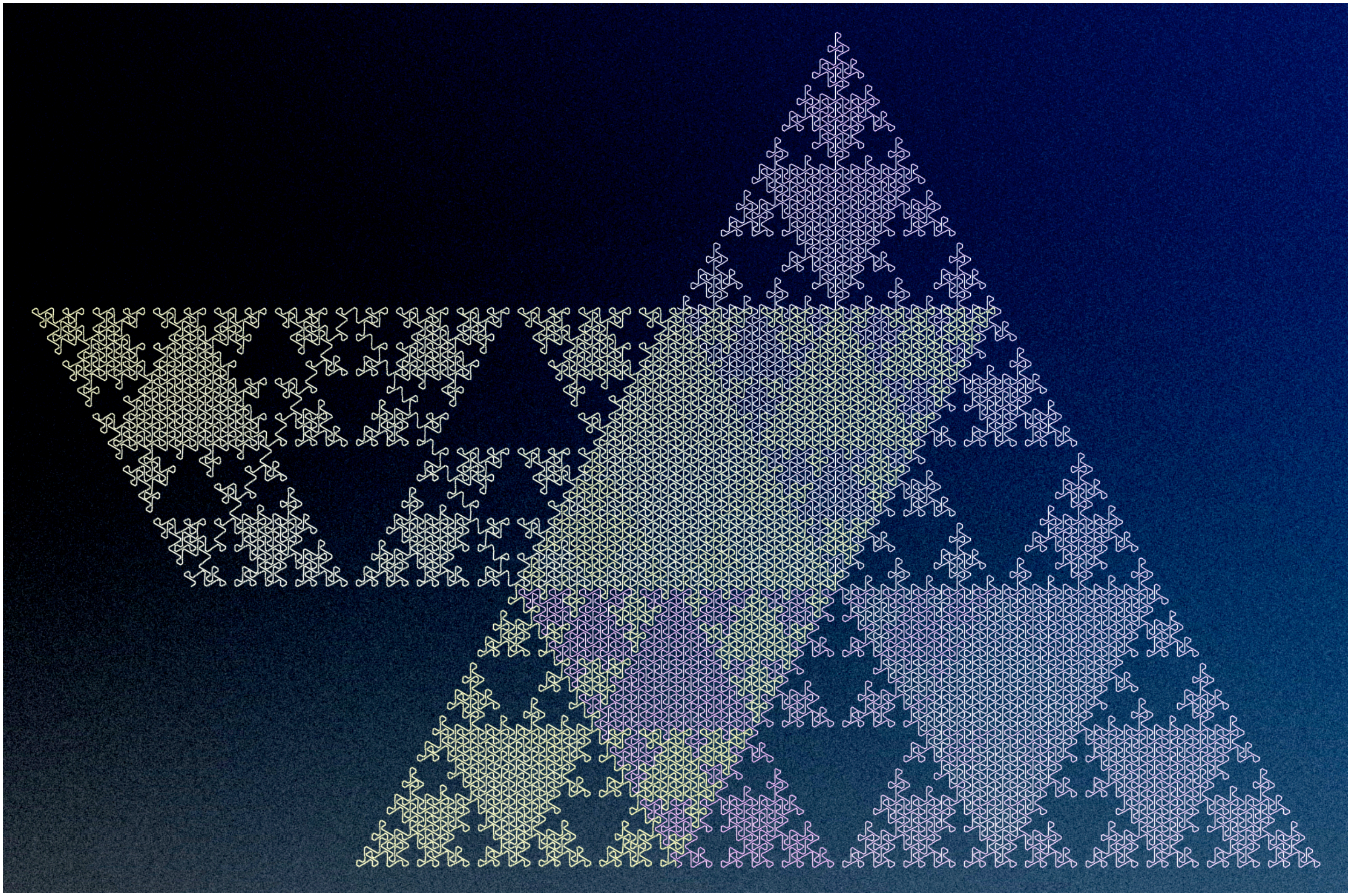
Here is that same generator with yet another set of flippings.



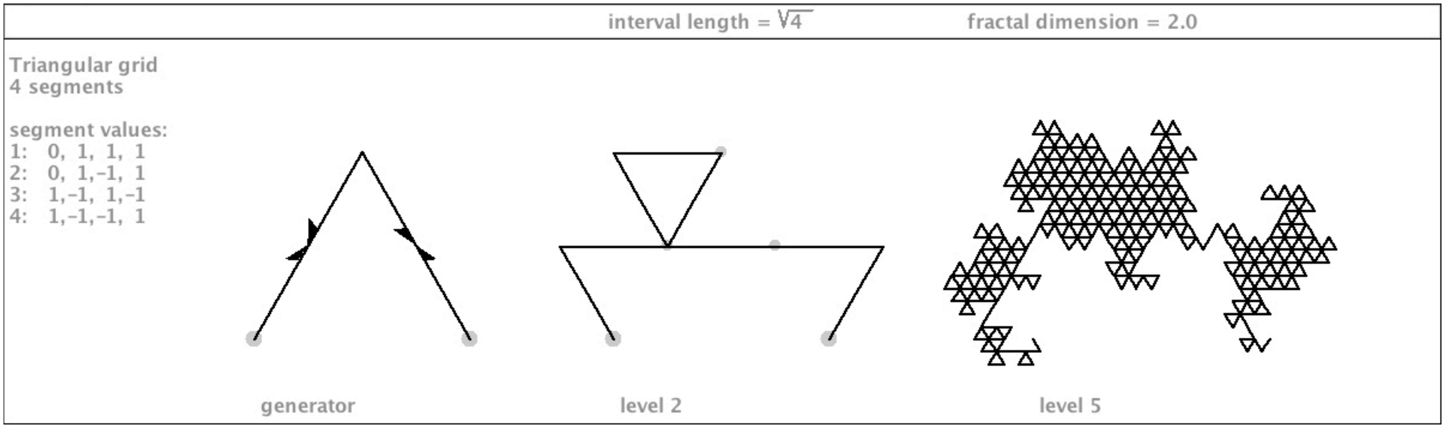
Now here is a different generator of this family. This curve is highly self-enveloping.

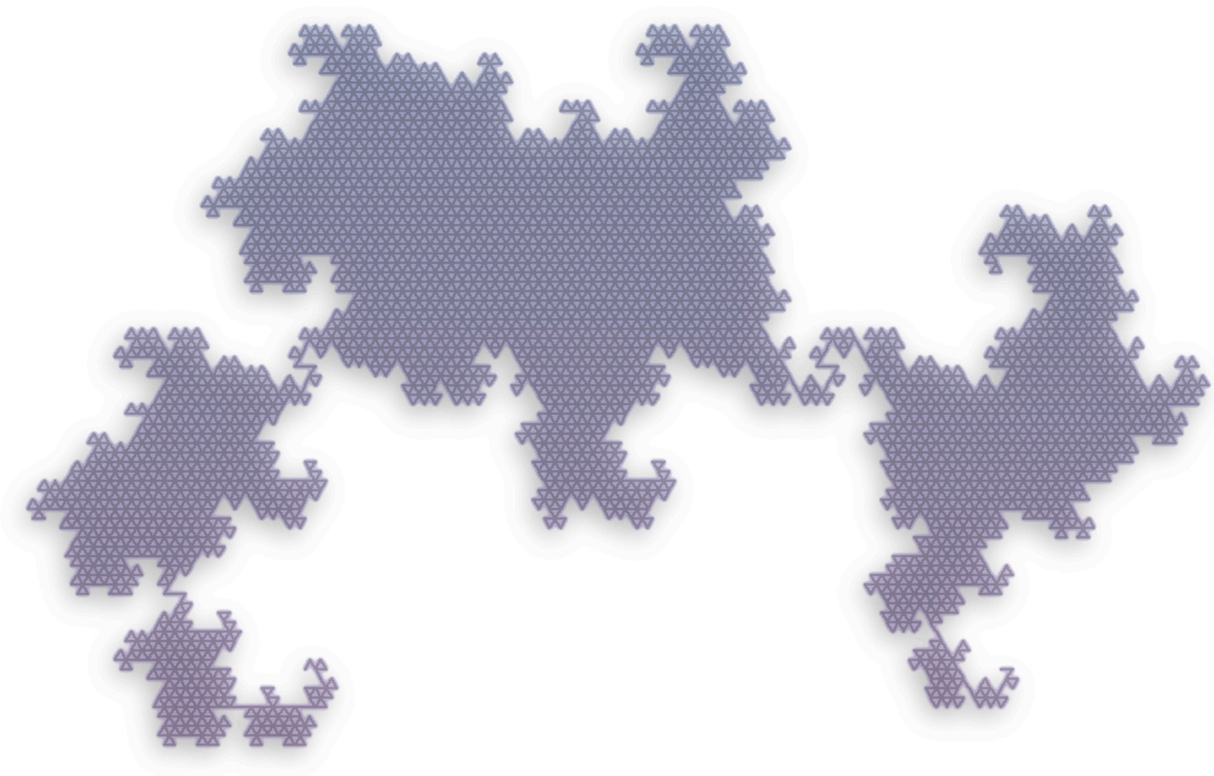


On the next page is a picture showing two copies of this curve…mating. They are engaged in the most intimate embrace one can imagine.

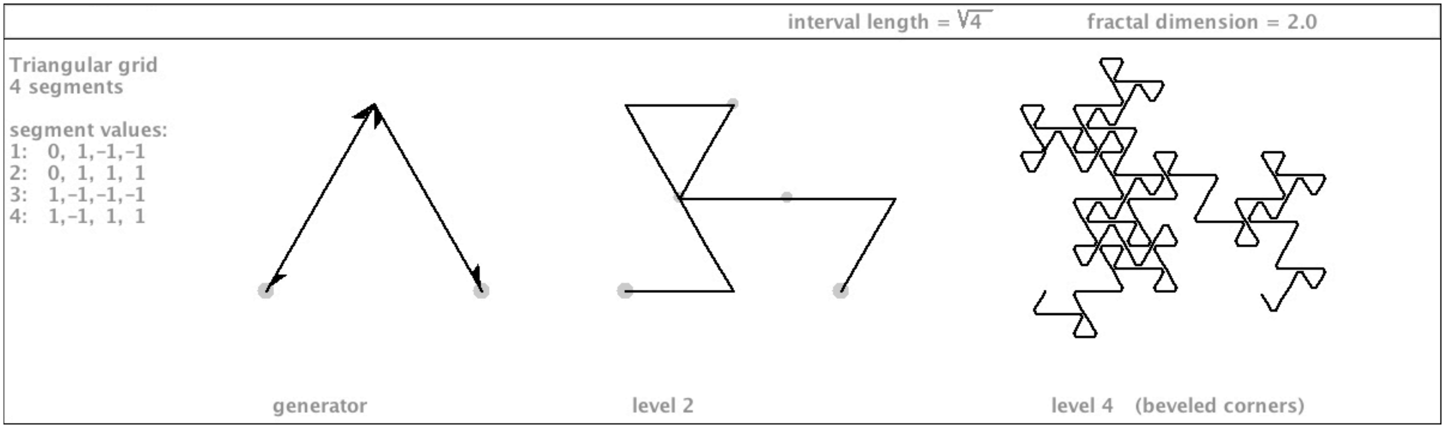


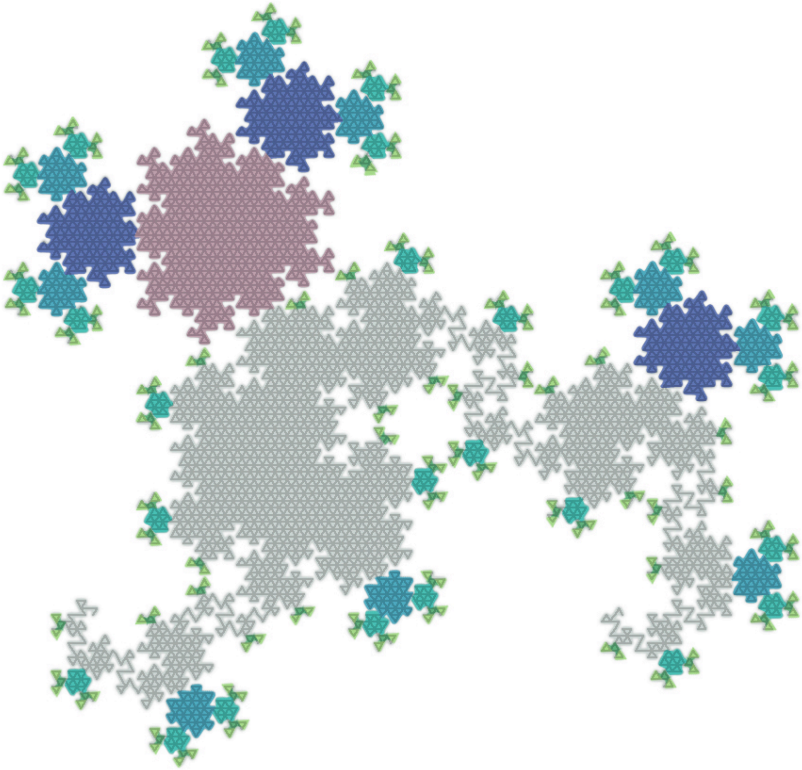
Here is the same generator shape with alternate flippings. This curve is very different. Every family has its token dragon curve – some of them are more dragon-like than others. Would you call this a dragon…or a scorpion? Or a….?



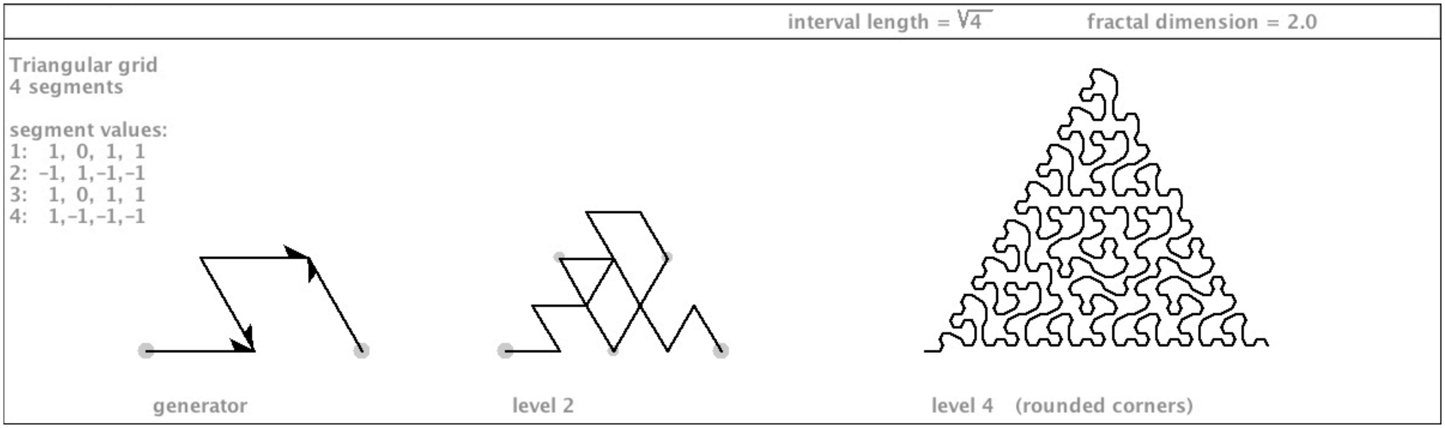


Here is another curve based on the same generator shape. I have rendered it below with hierarchical coloring to indicate the way nodes are formed: each node has 3-way symmetry and is connected to other nodes at pinch-points.

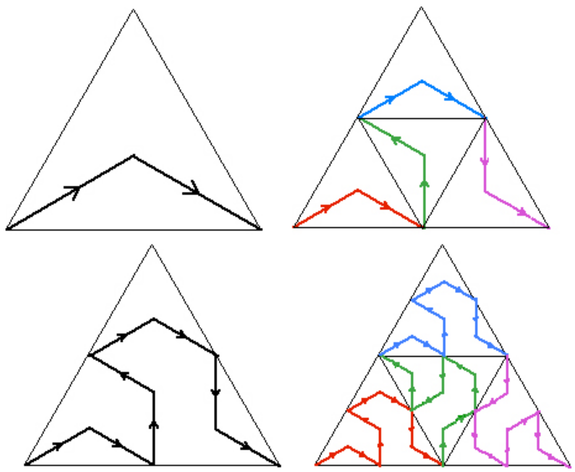


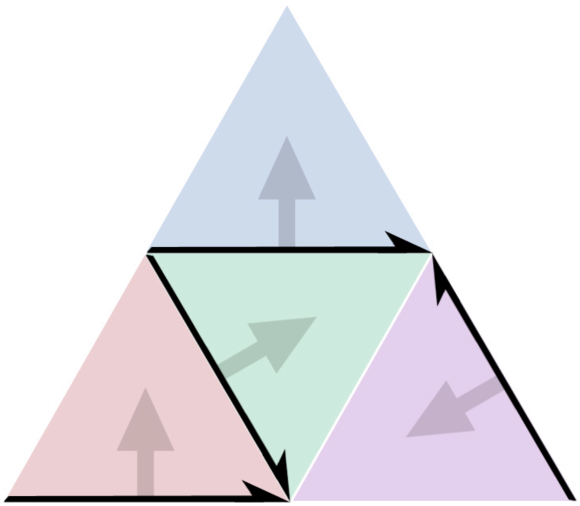


The last members of the √4 triangle grid family I will show you are curves that exactly fill an equilateral triangle. They are pseudo-gridfillers.

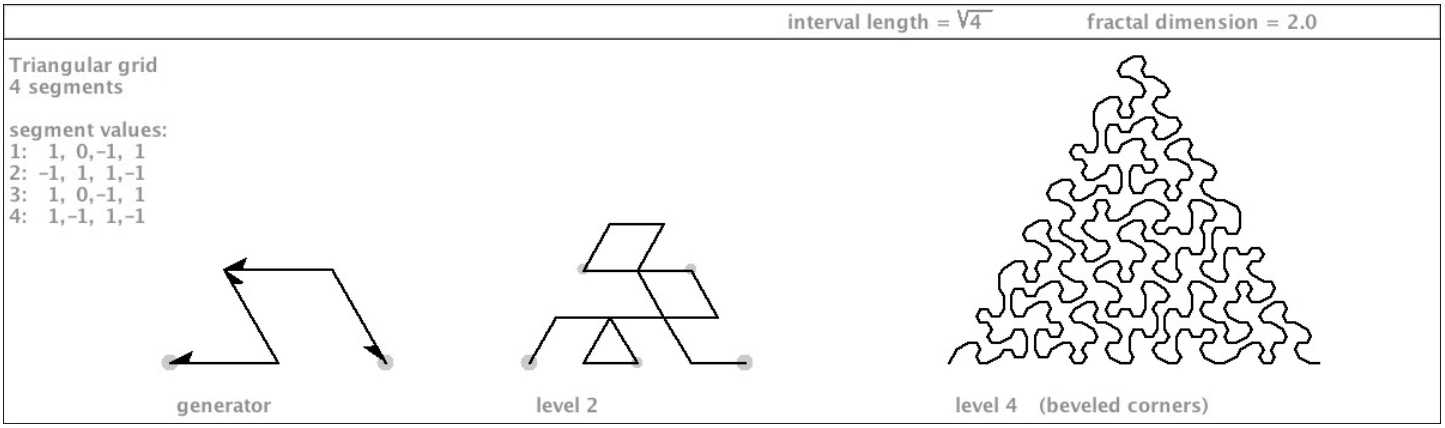


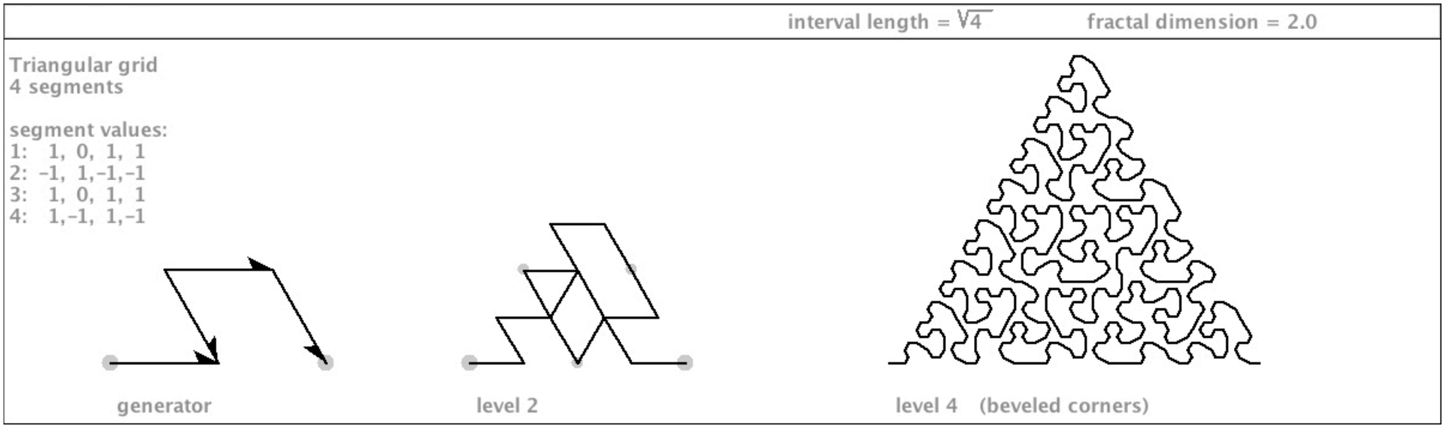
You can think of each of the four segments of this generator as being responsible for one of four sub-triangles. The precise set of segment flippings is important, so as to avoid edge-touching. Zbigniew Fiedorowicz [6] made a variation of this fractal – shown below.

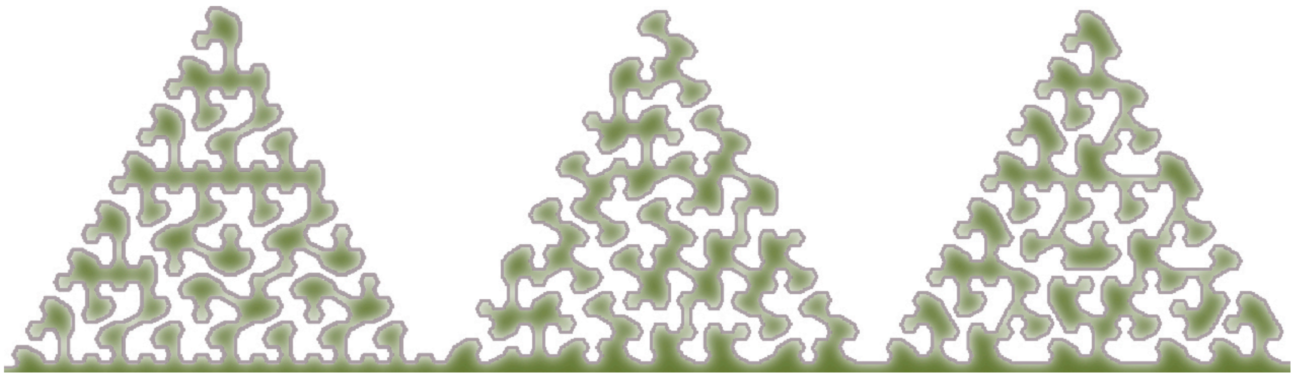




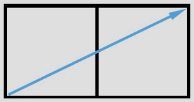
Two variations are shown here, and at the bottom of the page is a portrait of all three variations at level 4.



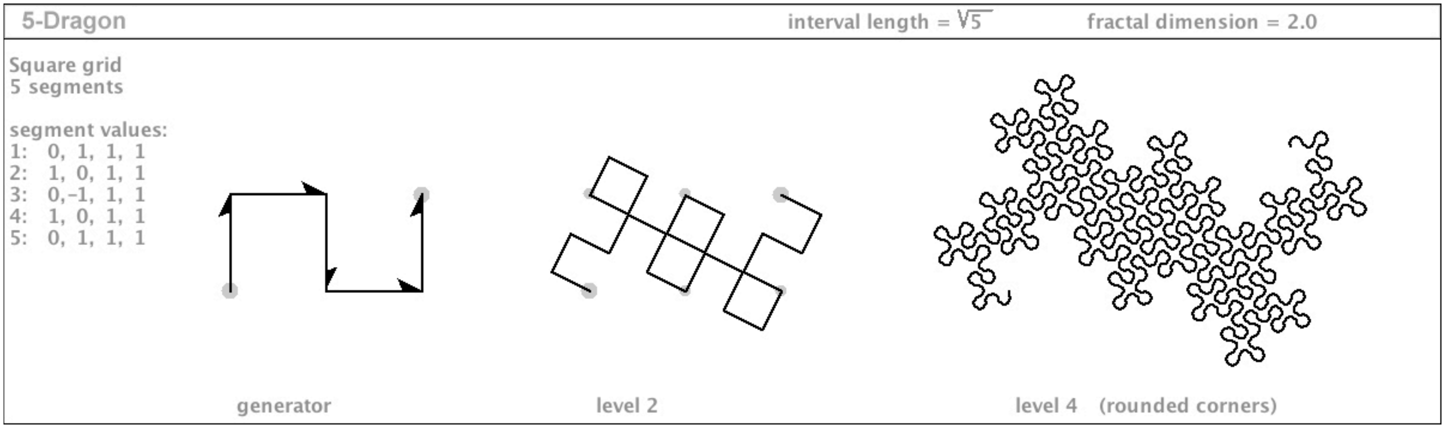




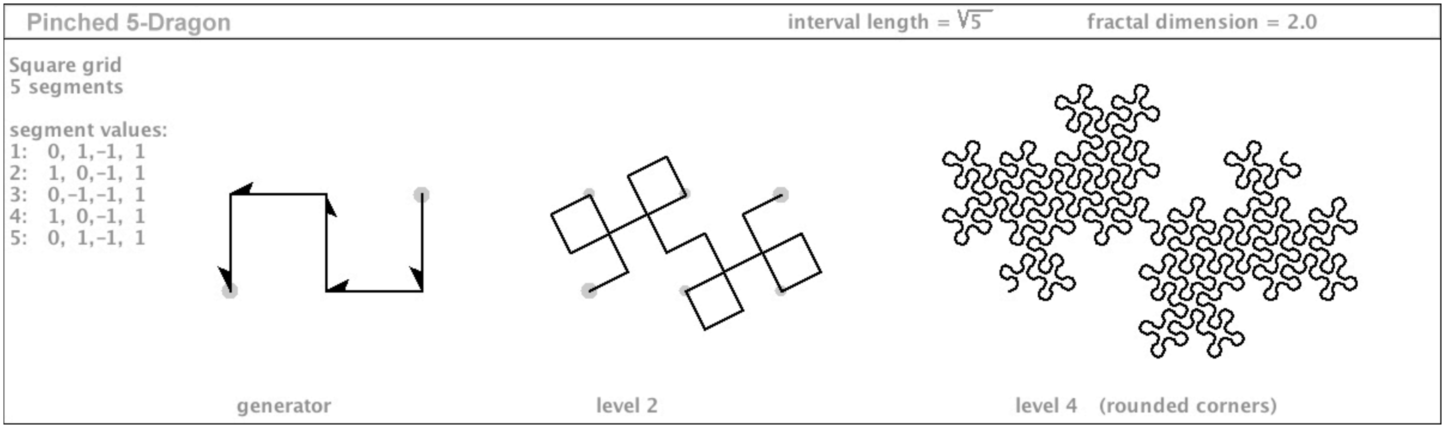
√5



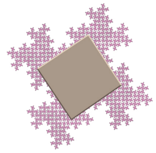
Now we come to the √5 family. We have already met the 5-dragon:

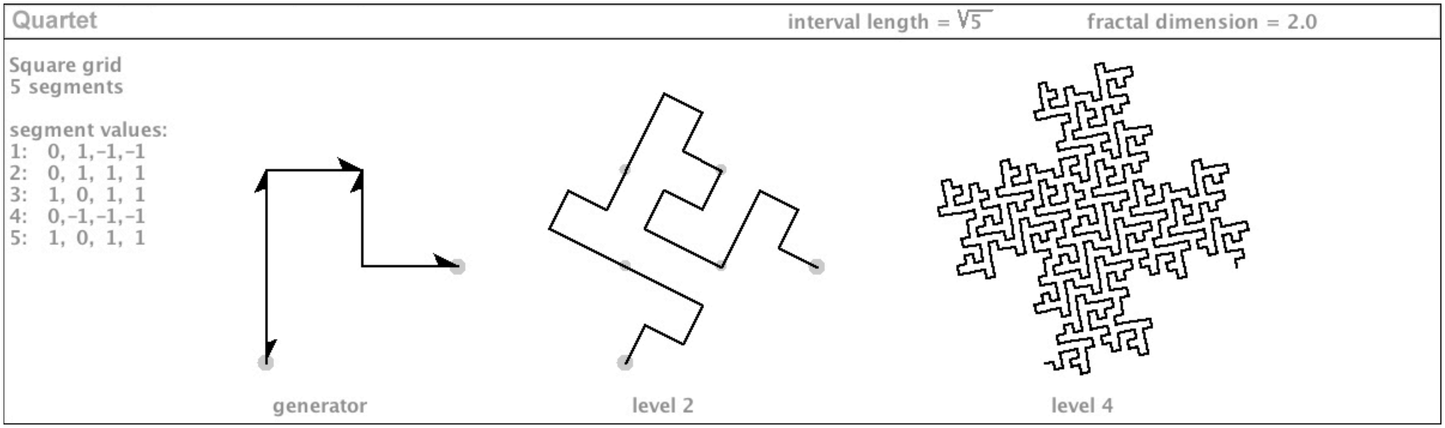


Remember how I flipped the x values of all the segments of the Ter-Dragon to make the *Inverted Ter-Dragon*? Well, the same can be done with the 5-Dragon. And, just like the inverted Ter, the inverted 5-Dragon has a pinched waist.

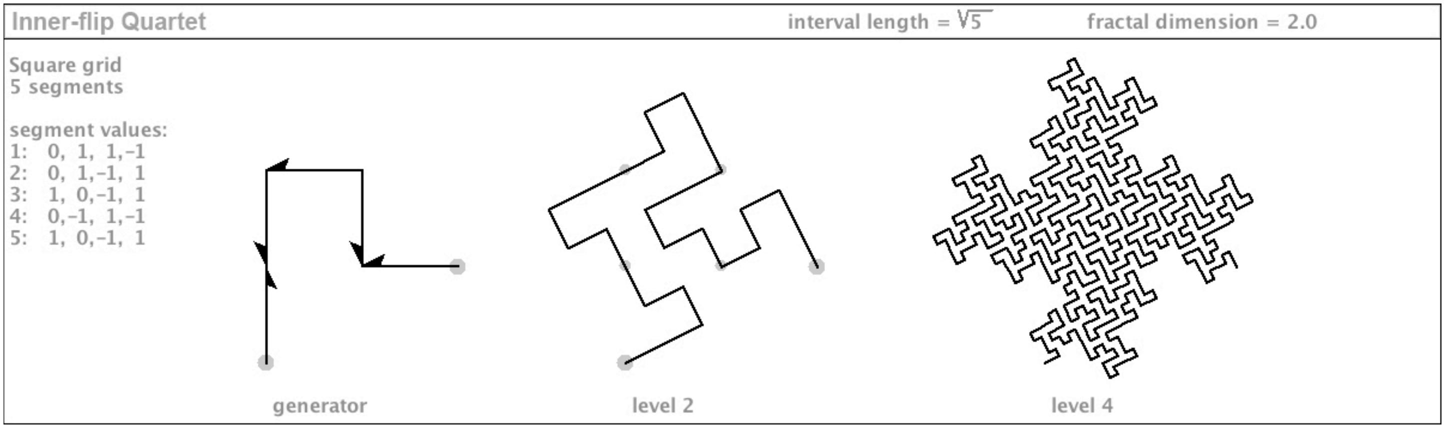


This next specimen is Mandelbrot's *Quartet*: "Each 'player', and the table between them, pertile." [16]. He claims to have "designed" it, although one could debate that such a curve is "discovered" rather than "designed". In either case, it is one of the finest self-avoiders.

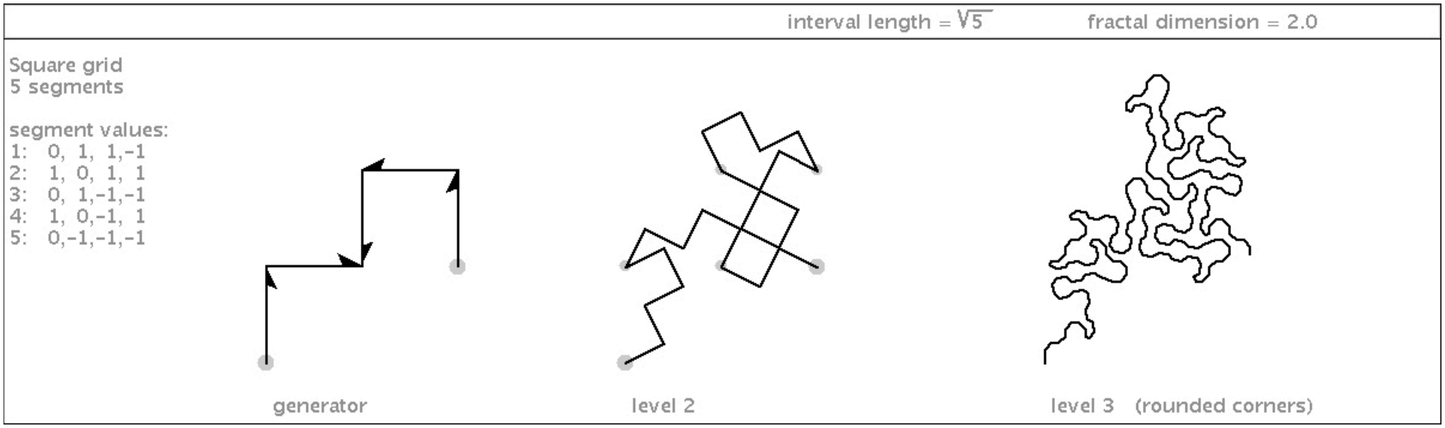




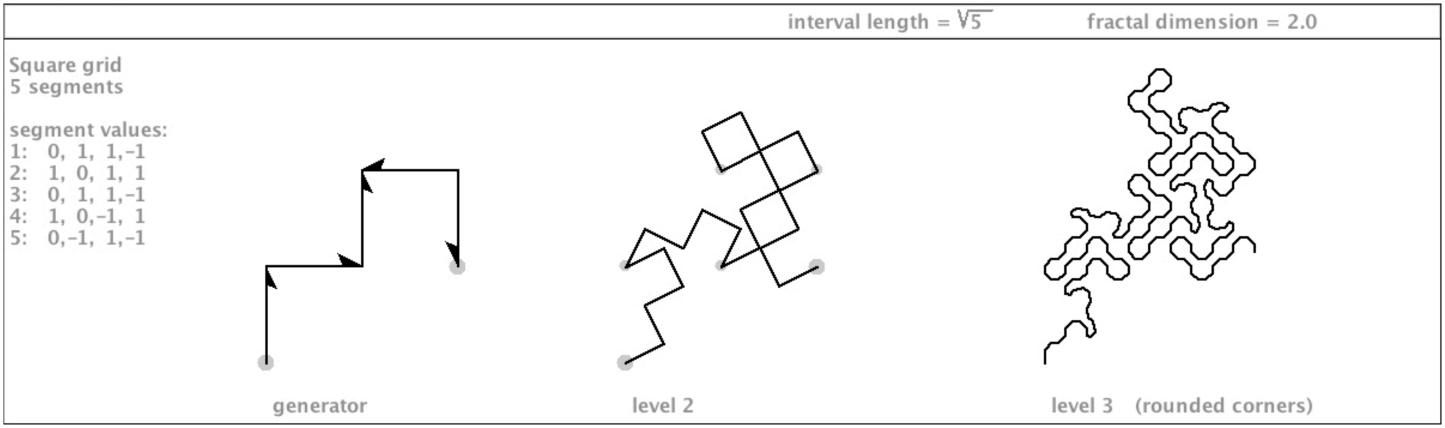
I discovered a variation of this generator, created by reversing the x-flipping of each segment. I call it "Inner-flip Quartet".



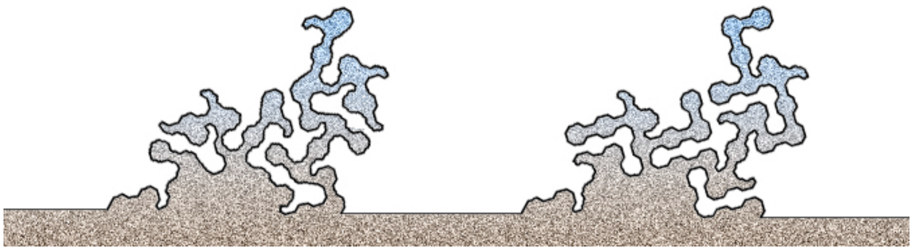
Next I will show you six variations on a single generator shape. Two examples are shown on the next page. The first example has an interesting property: due to the flippings, the orientations of copies of the generator do not correspond with a continuous square grid. You can see this in the mixture of 90 and 45-degree angles in the level 2 teragon. I would not have expected a curve like this to survive the fractal test. There is indeed self-contacting in several vertices, but other than that, it is rather well-behaved, as indicated by rendering with rounded corners.



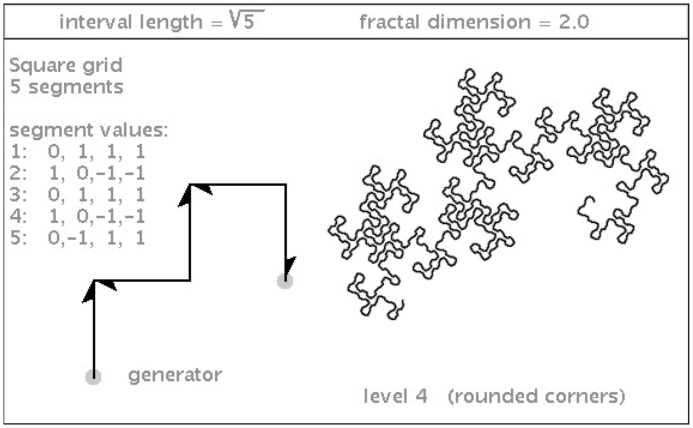
A close relative of this curve is shown here.

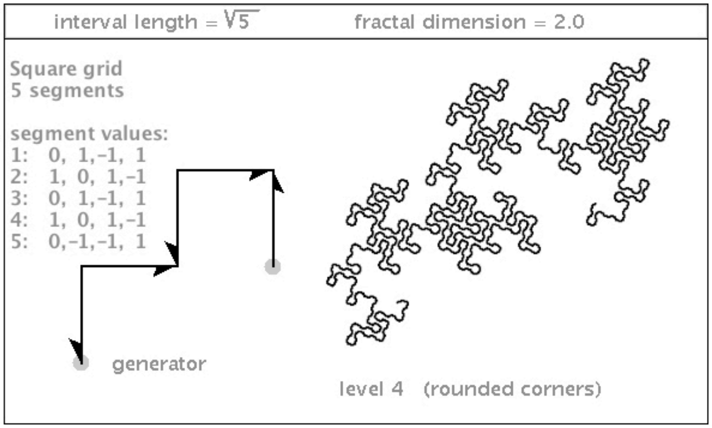


These two specimens resolve to the same general shape, as indicated by the illustration below.

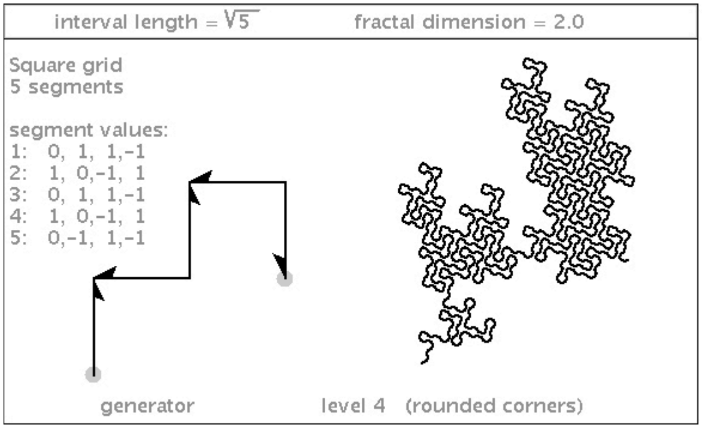


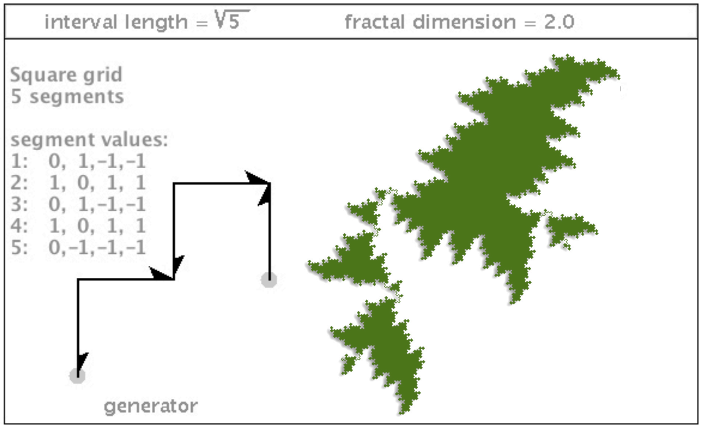
Given the same generator, with alternate flippings, we get two gridfillers with very craggy boundaries:





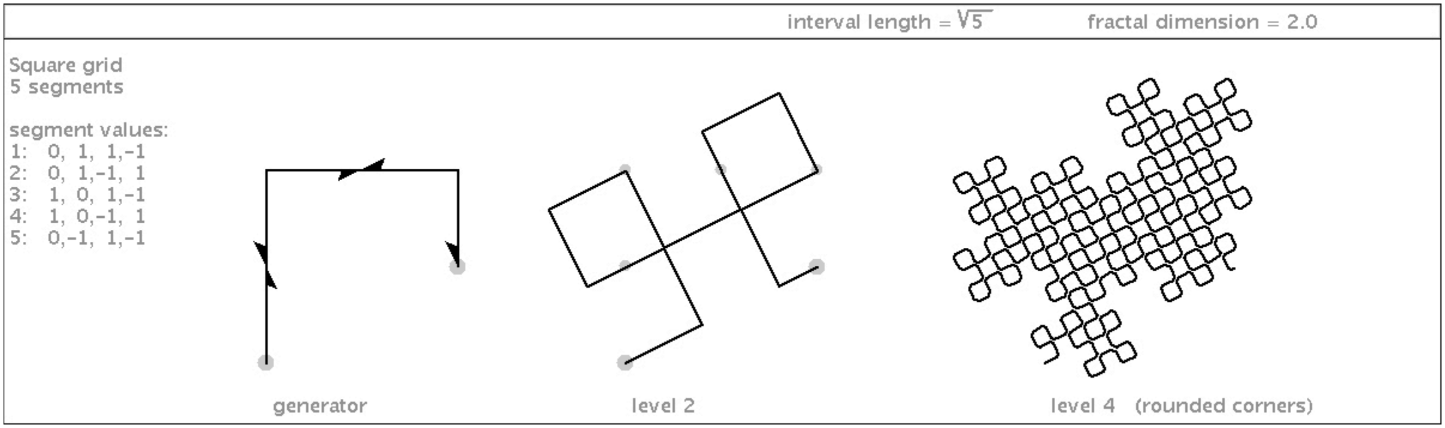
With other changes in flippings, we get the following gridfillers:

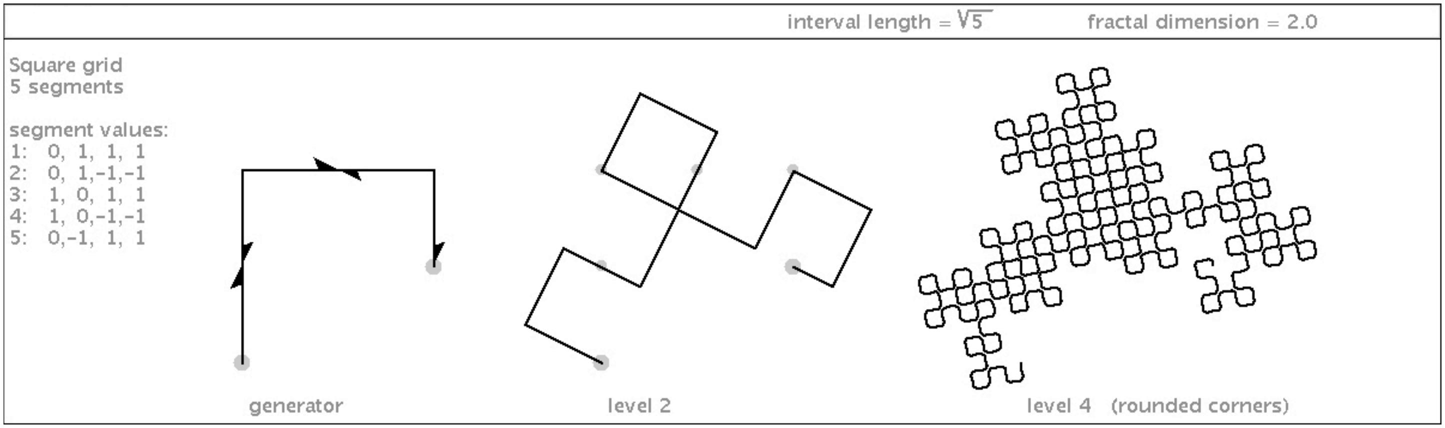




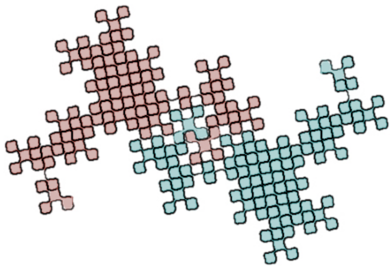
In that last one, notice how the conifer tree-like spike at the upper-right corresponds to the empty gap at the bottom, rotated by 90 degrees. My brain is pertiling!

Here are two plane-filling curves of the √5 family that use a common generator shape.

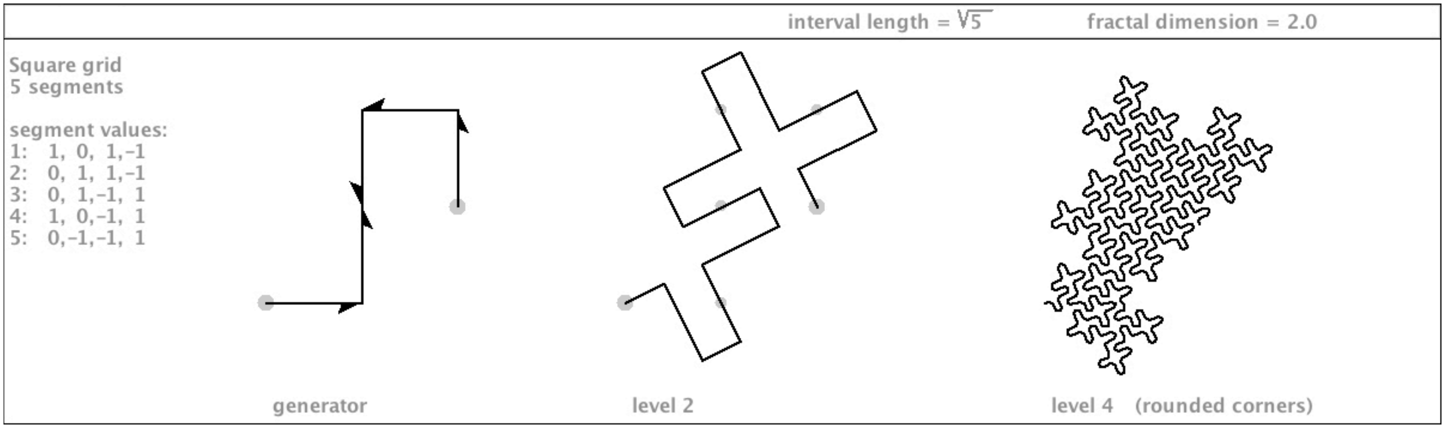


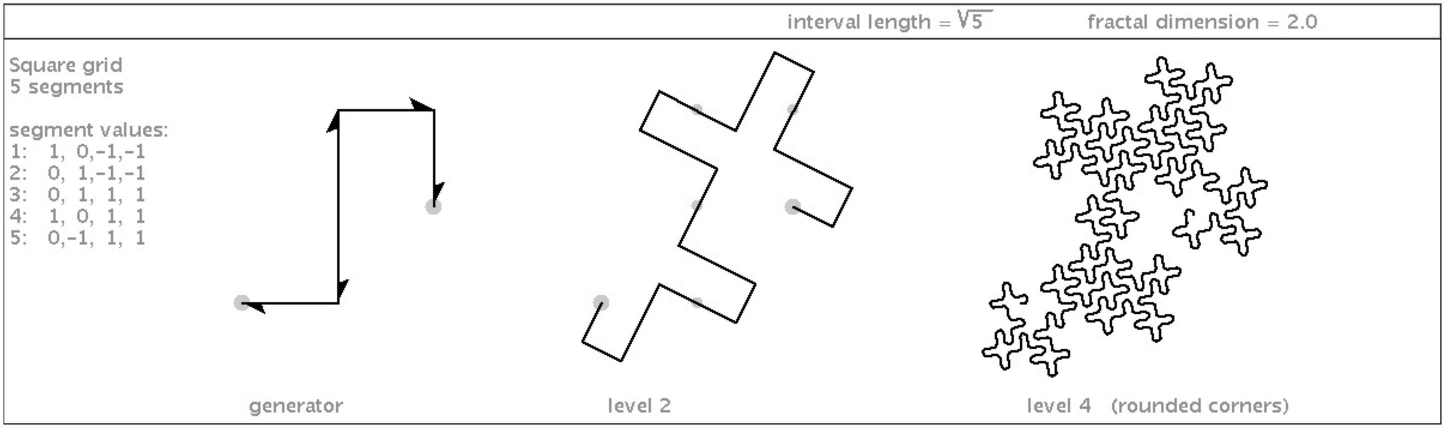


That last curve can be combined with a 180-degree flipped copy of itself to make the shape of the 5-Dragon…

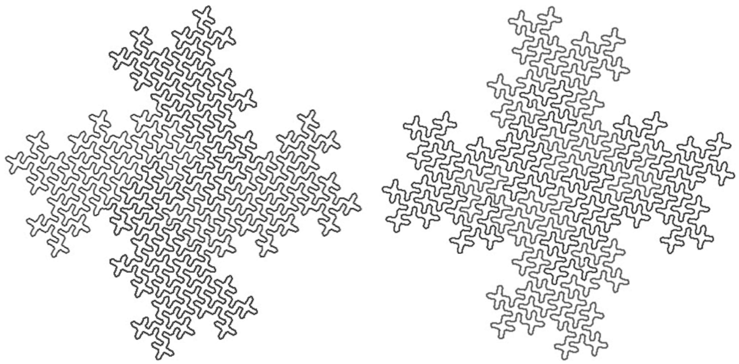


This next curve is a self-avoider. It is followed by a similar specimen.



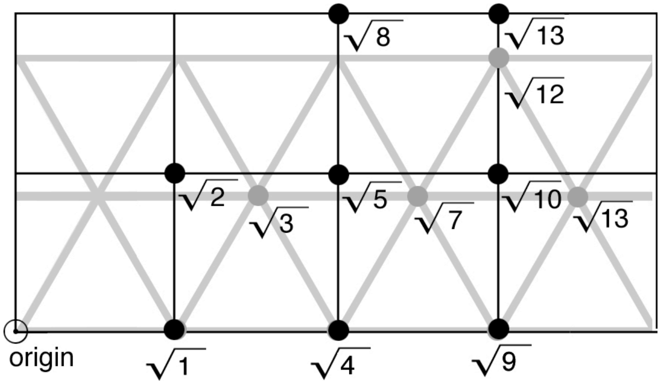


Each of these last two curves can be copied four times – each copy rotated 90 degrees – and joined together to make a continuous curve. The overall shape is a replica of the Quartet (one of them is a mirror-image of the other). This appears to be a property of many √5 curves I have shown.



√6 ?

There are no √6 plane-filling curves within my scheme! Why? Well, I can tell you this: it has something to do with the grids. Here's that illustration I showed you earlier:

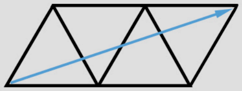


An uninspiring answer to the question (why no √6?) is that there are no grid points on either a square or triangular grid whose distance to the origin is √6. I could just leave it at that, and say "let's move on", but I seek a deeper answer. Notice also that there is no square root of 11 distance either. There is also no square root of 14 distance. The list continues in a way that is reminiscent of the erratic series of prime numbers.

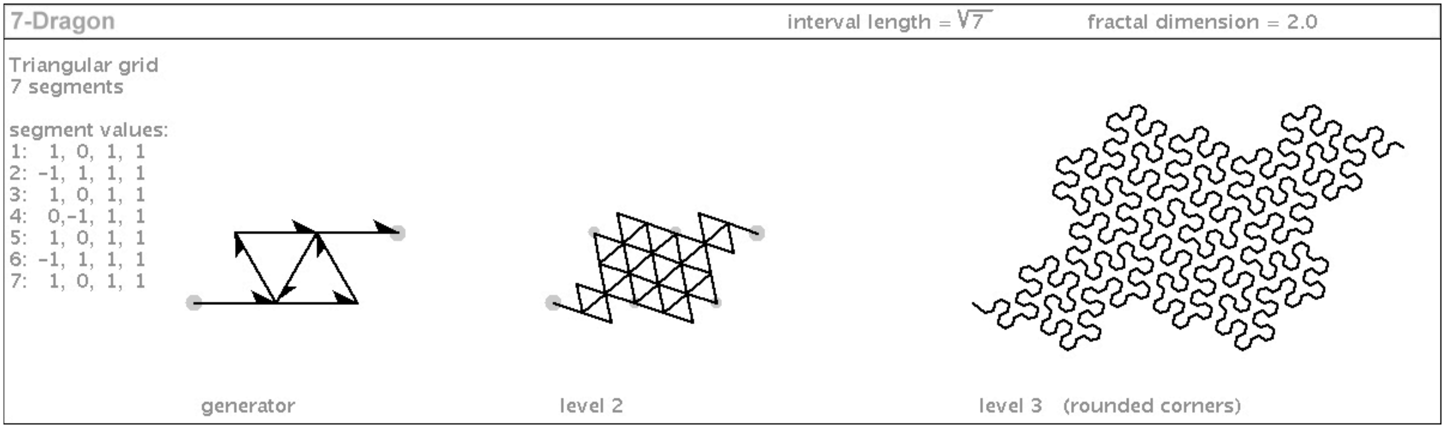
In the case of the square grid, the answer is simple: each of these distances is the sum of two squares. Look no further than the Pythagorean theorem to see why this is so. But when considering the triangle grid, it is a little less obvious why we should end up with the set: √1, √3, √4, √7, √12, √13, etc. I shall leave this question open for you to explore on your own. Also, did you notice? …there are two √13 distances – one on the square grid and one on the triangular grid. What's up with that?

Now, I must admit: earlier I claimed that all plane-filling curves have interval lengths that fall between grid points of either the square or triangular grids. But I cannot say for sure that this is true. It may be that any number of generators with arbitrary interval lengths can yield plane-filling curves (although they may evade simple mathematical analysis). I leave it up to you, dear reader/viewer/thinker, to give me an inspiring answer. You can always find me at Jeffrey@Ventrella.com. Okay, I'm afraid I am going to have to say, "let's move on now"…to the awesome √7 family.

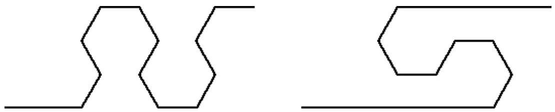
$\sqrt{7}$

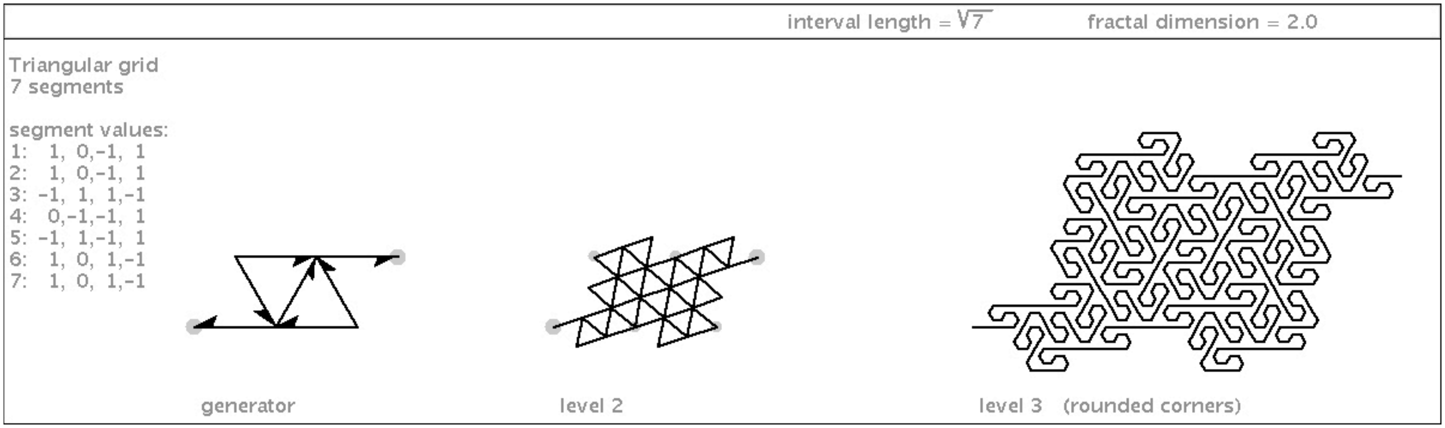


Now we can add a new specimen to the Ter Dragon and the 5-Dragon to make a nice neat trio of prime number palindrome dragons. It is the *7-Dragon*. These three curves have a few things in common: (1) they are all palindromes; (2) they require no segment flippings; (3) they represent the first three odd numbers (other than 1); and (4) they are all gridfillers. One difference is that the 7-Dragon is rather fat compared to the other two, and it is not as rough around the edges, so it may not qualify as a true “dragon”. It also cannot be converted into a pinched-waist specimen by way of flipping the x values of its generator segments. Clearly, the 7-Dragon has a bit of a weight problem. Here it is:

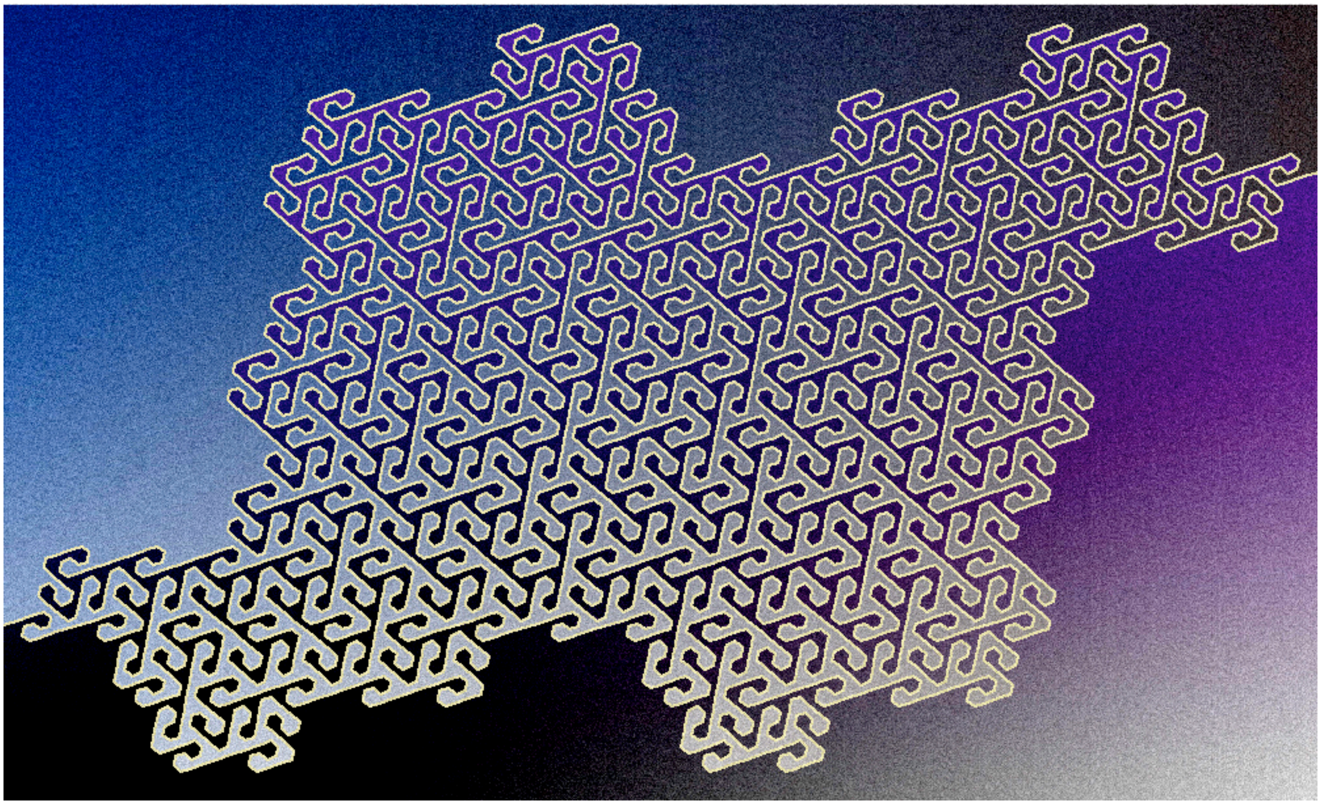


I found a variation on the 7-Dragon generator that traverses the same segments, only in a different order. The diagram at right reveals this difference by way of rounding the corners of the generator. This alternate 7-Dragon is shown on the next page.

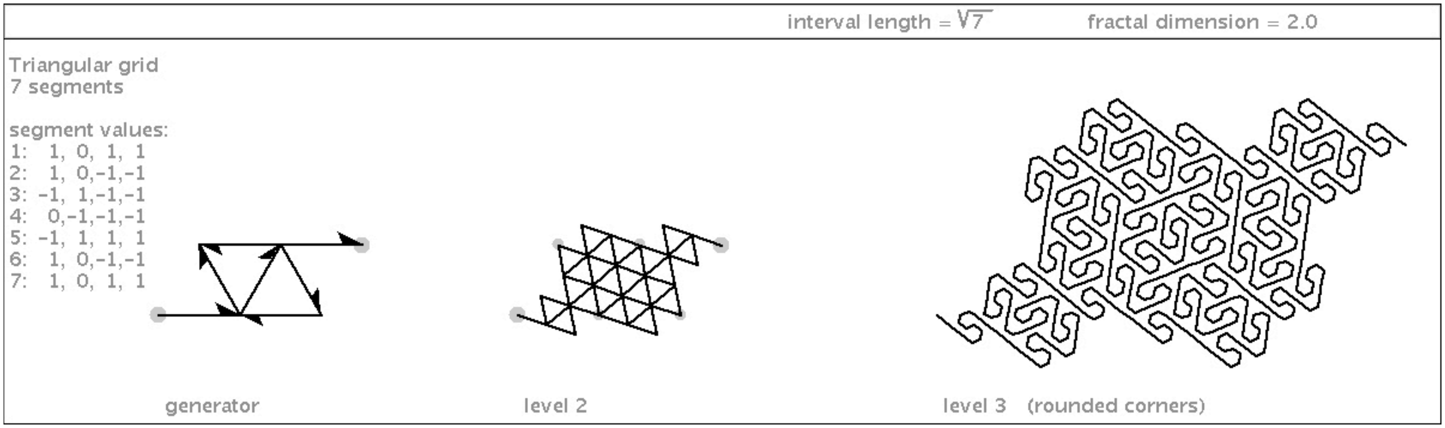




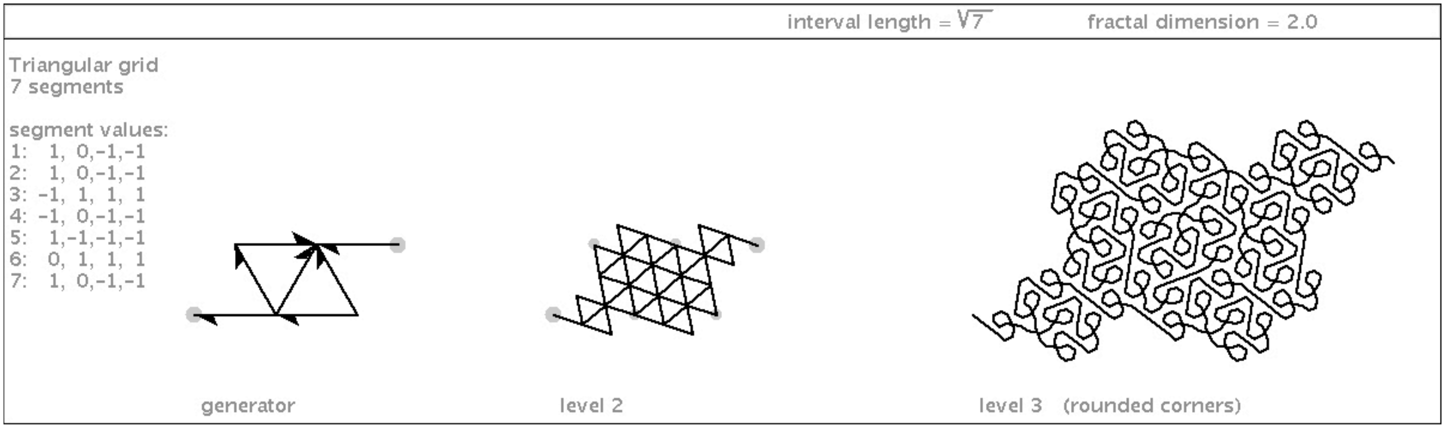
Here is a color rendering, with rounded corners, showing how the curve is a boundary between two domains:



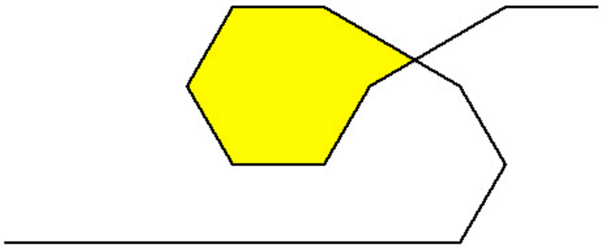
Here is yet another variation on the 7-Dragon:

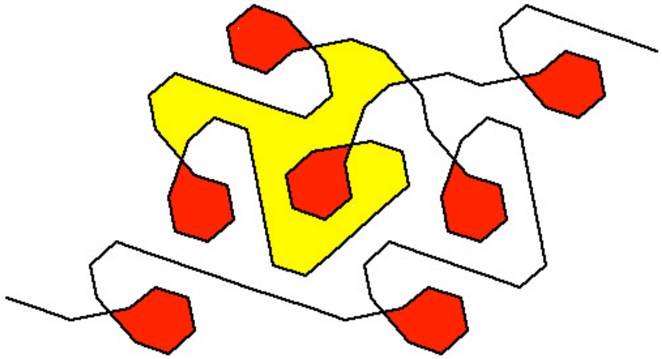


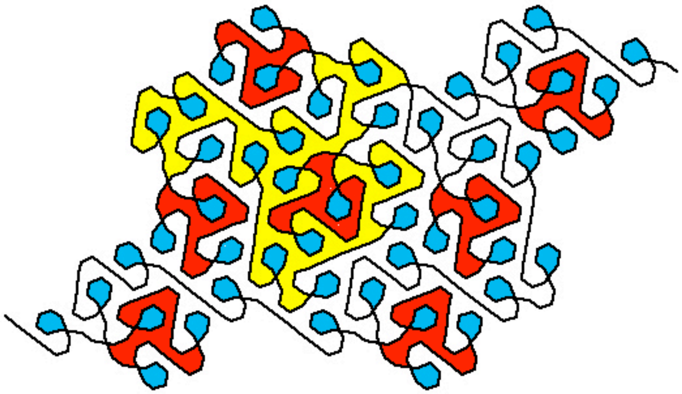
Now, here is a strange specimen. I wouldn't normally include a self-crossing fractal curve in this book, but this variation on the 7-Dragon is irresistibly clever in it self-crossing:



On the next page is a diagram that shows the progression of this fractal curve. As usual for a gridfiller (or, in this case a self-crossing gridfiller) I render it with rounded corners. For this illustration, I filled-in parts of it with color, to show how regions percolate into ever-complexifying domains.

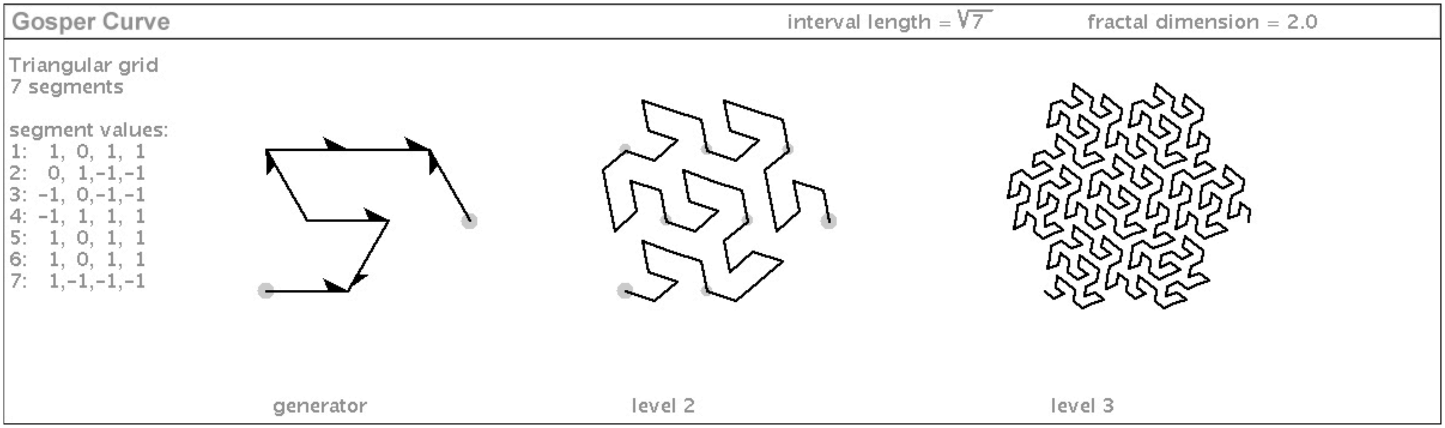


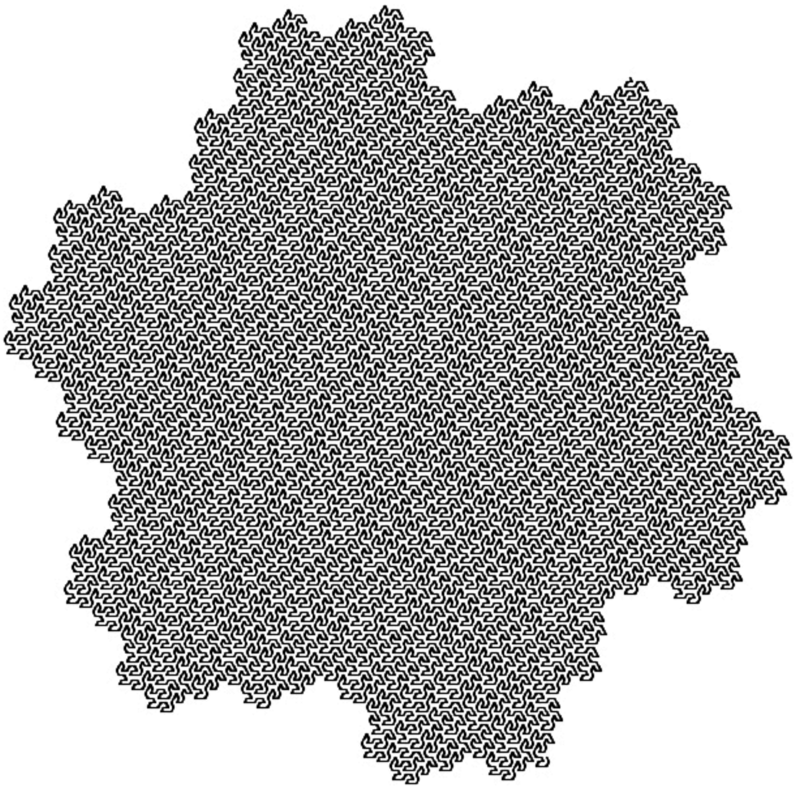




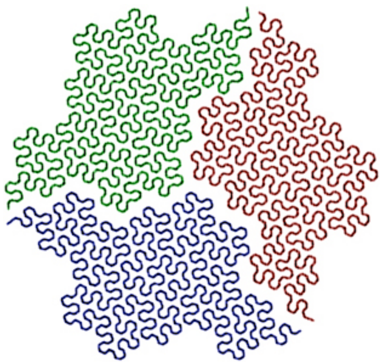
Notice how the crossings accumulate with each level. In level 1 (upper-left) there is one crossing, which forms a closed region shown in yellow. In level 2 (upper-right) there are 7 small-loop crossings that form closed regions colored in red, and one larger-scale crossing (leftover from level 1, shown in yellow). In level 3 (bottom) there are 49 small-looped crossings that form closed off regions colored in blue. The remaining yellow and red regions have become more convoluted. I would assume that with each continuing level of fractalization, the remaining regions would stay topologically whole yet more convoluted, and that the self-similarity of clusterings would continue at each level of detail.

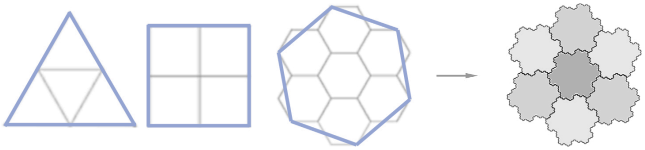
Now it is time to visit one of the most famous plane-filling curves of all: the *Gosper Curve*, a truly splendid self-avoider.





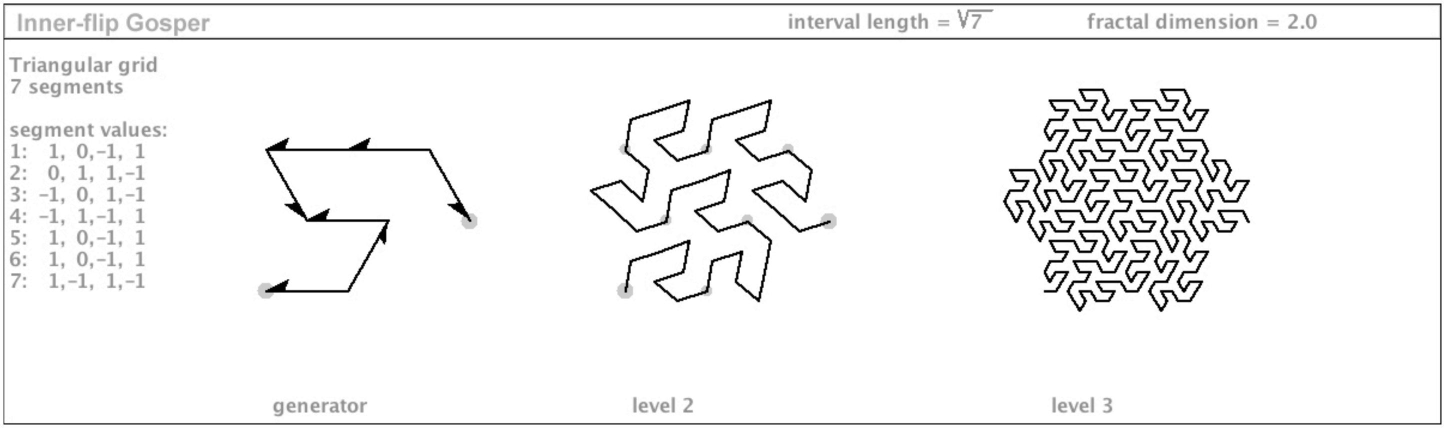
The Gosper curve is named after William Gosper [7]. It fills a roughly hexagonal region, which Mandelbrot called "Gosper Island". The shape of the Gosper Island will be popping up throughout our exploration of this family. Here's one example: three 7-Dragons can be combined to form a shape of the Gosper Island.



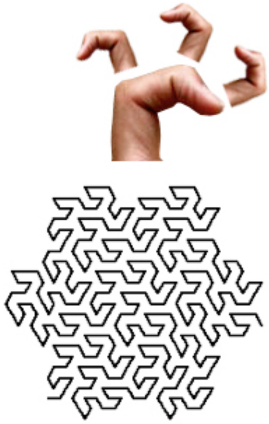


It is a source of frustration for geometers that hexagons do not tile recursively, like squares and triangles. In other words, they are not rep-tiles. But the Gosper Curve defines a series of seven tiling regions that are all similar to the whole. So it gets around this problem…that is, if you don't mind having hexagons with craggy boundaries.

Here is something I call the "Inner-flip Gosper". It has each of the x-values flipped, and so it creates a different character to the way the fingers curl to fill up the Gosper Island. Counting up the fractal levels, the direction of the curl reverses.

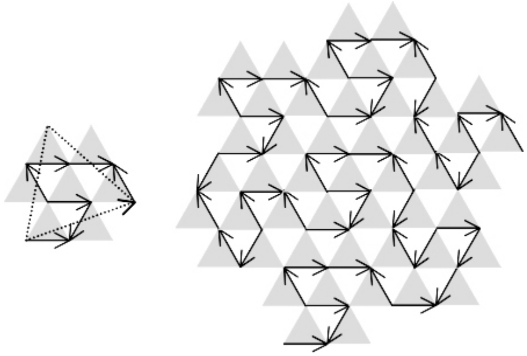


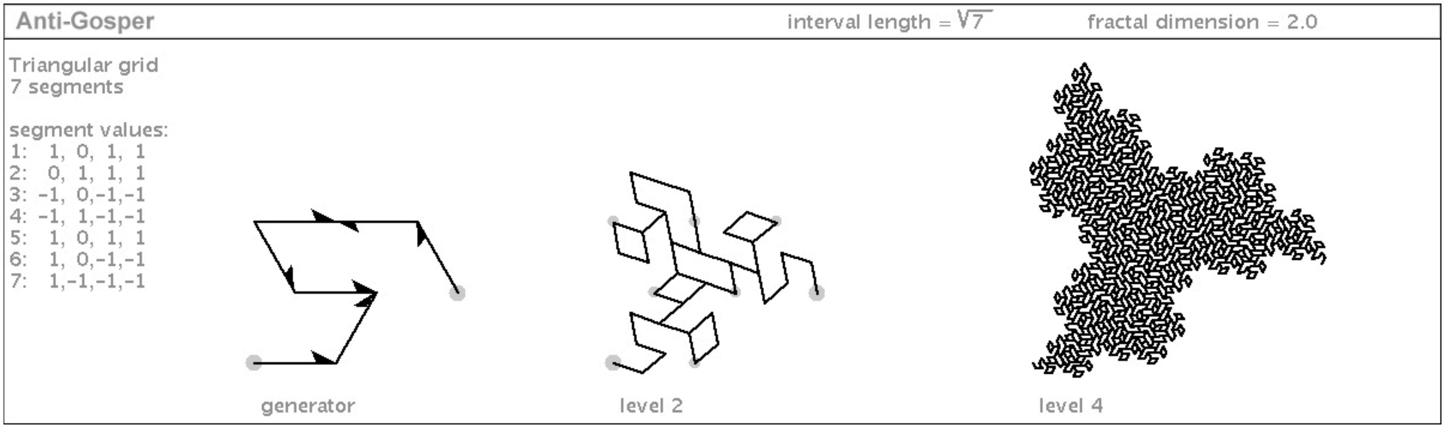
Gosper



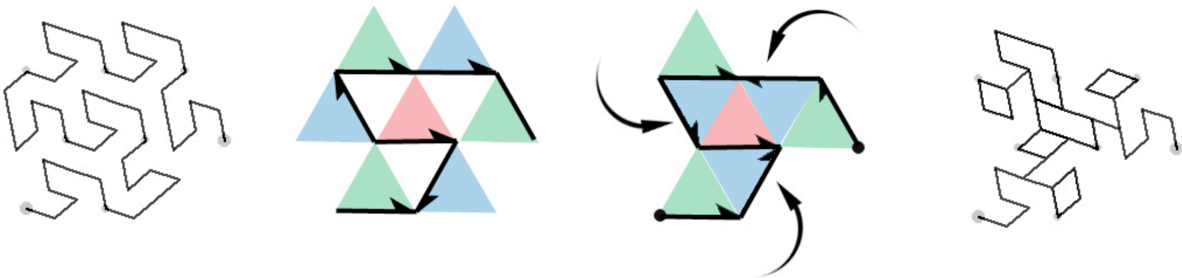
Inner-flip Gosper

The diagram at the right was created by Fukuda, et. al [4] to show a generalized scheme for constructing Gosper-like Curves. It can be described as a triangular checkerboard, where each gray triangle stands for a hexagonal tile. Now, keep these triangles in your mind as I show you a variation of the Gosper Curve's generator that I discovered. I call it "Anti-Gosper". Normally, the Gosper Curve is a happy self-avoiding curve. But I have seen the Gosper Curve when it is in a bad mood. When this happens, it flips three of its segments, causing its fractalized teragon to become a shriveled-up triangle. The Anti-Gosper is *edge-touching*, in a rather anti-social way.



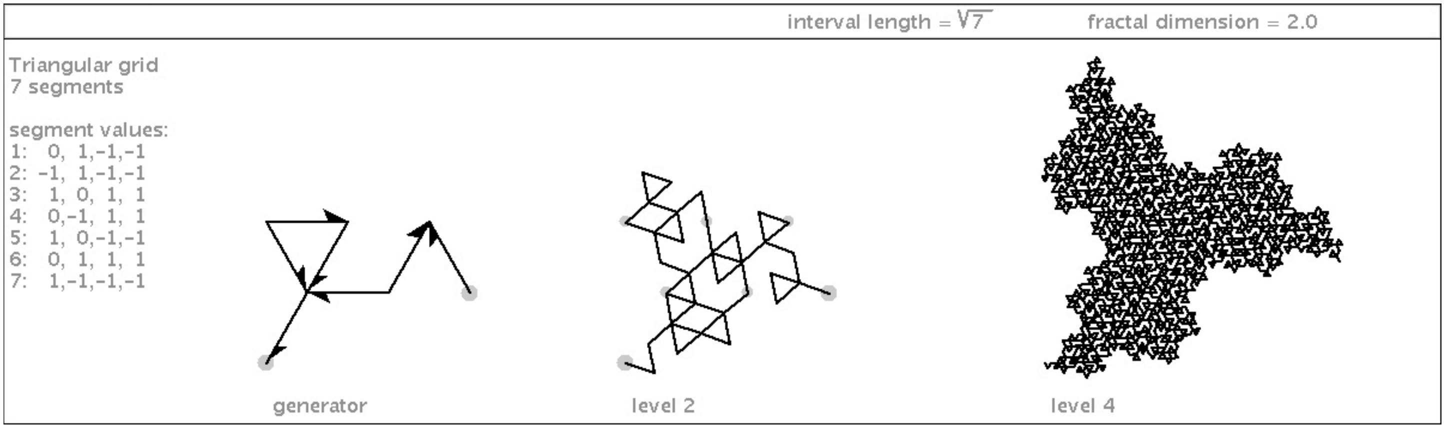


Here is a diagram illustrating the transformation from Gosper to Anti-Gosper, where the checkerboard of triangles collapses in on itself, leaving no gaps.

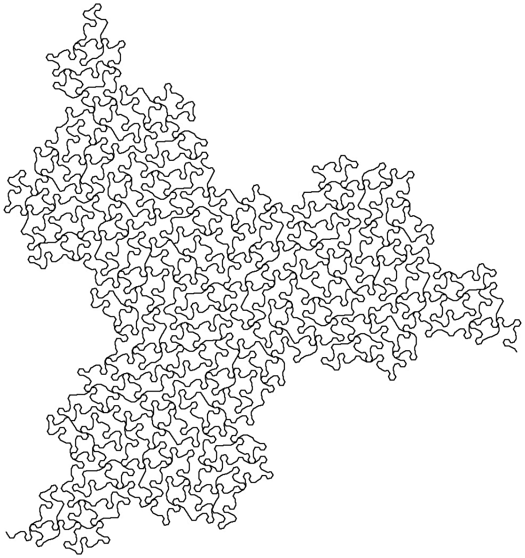


The Anti-Gosper essentially transforms the Gosper curve from a hexagon-like tiling fractal to a triangle-like tiling fractal.

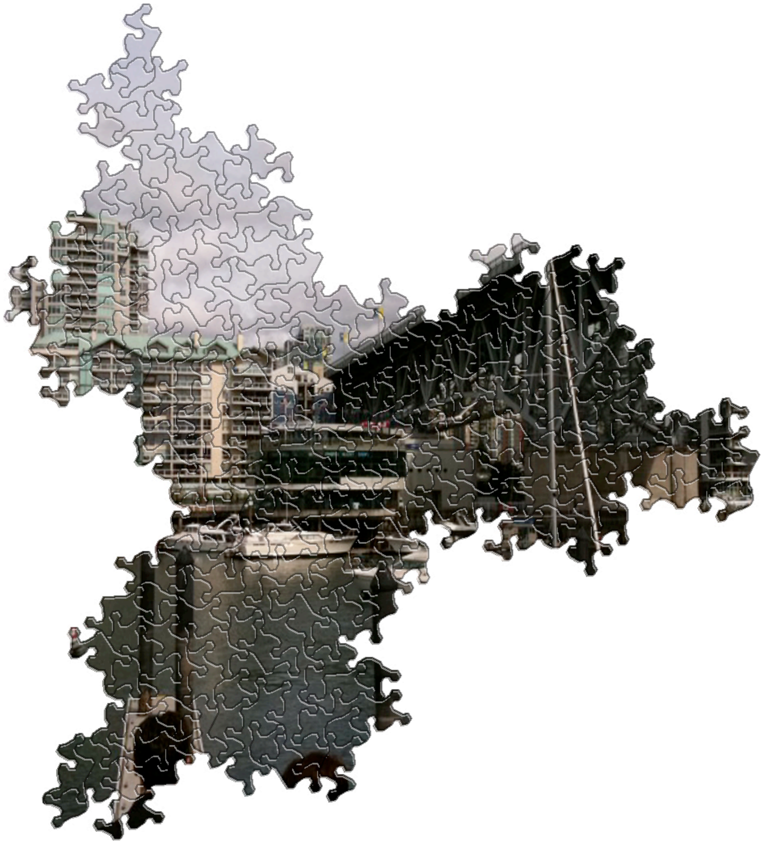
The overall shape of the anti-Gosper can be created by another generator, shown here:



This specimen is shown below with rounded corners. You can easily see that this plane-filling curve is self-contacting in such a way that rounding the corners is not sufficient to separate-out the touching parts.



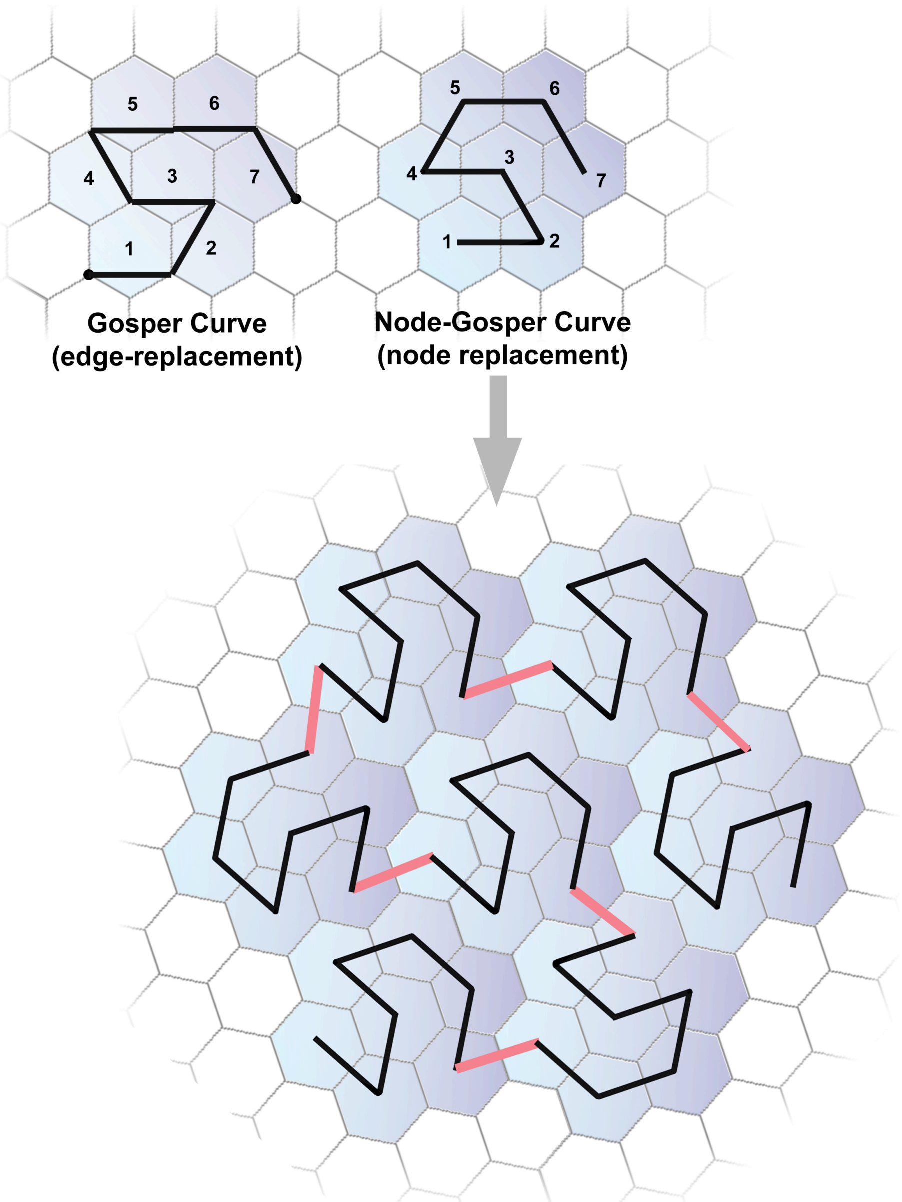
However, it does make a fine jigsaw puzzle :)



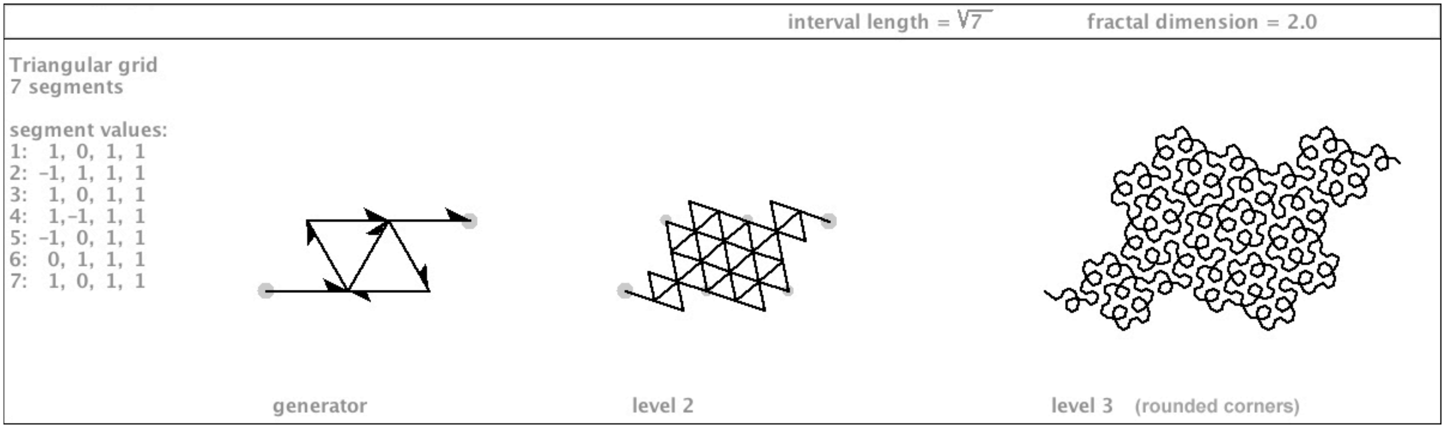
Node Gosper

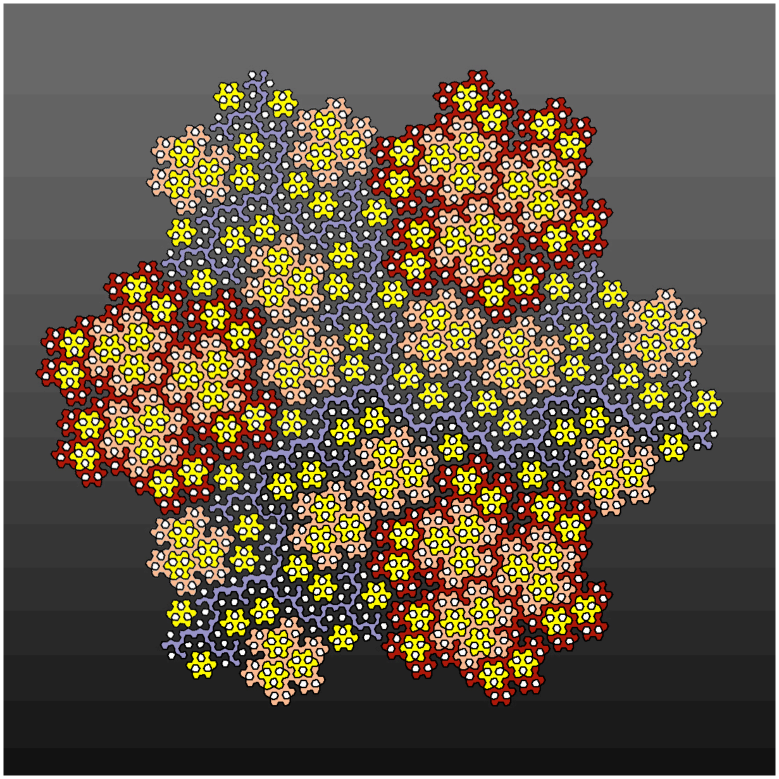
Remember the node-replacement curves I showed you earlier? Well, after discovering that any fractal tiling could be used as the basis for a node-replacement curve, I realized that the Gosper curve would qualify. So here is a picture I drew of a variant of the Gosper curve, which I call "Node Gosper". Instead of the segments of the generator spanning between two corners of each hexagonal tile, they connect at midpoints of neighboring hexagonal tiles.

Just for fun, I did a drawing of the second teragon of a node Gosper, and rendered the connective tissue with pink lines. Did you notice that fractal curves generated with node replacement are not strictly self-similar? The extra connective lines create slight differences in the internal shapes. In this example you can see the slight variations among the bumps. These variations accumulate progressively with each teragon.

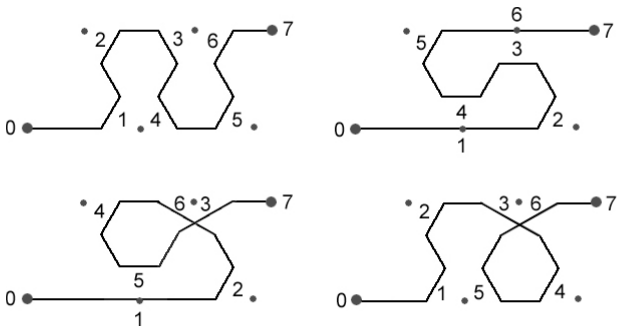


The √7 family is genetically imbued with a talent for clever self-crossing. Let me show you another variation of a self-crossing 7-dragon, followed by a colorful self-crossing Gosper Island made by combining three copies (below at left).

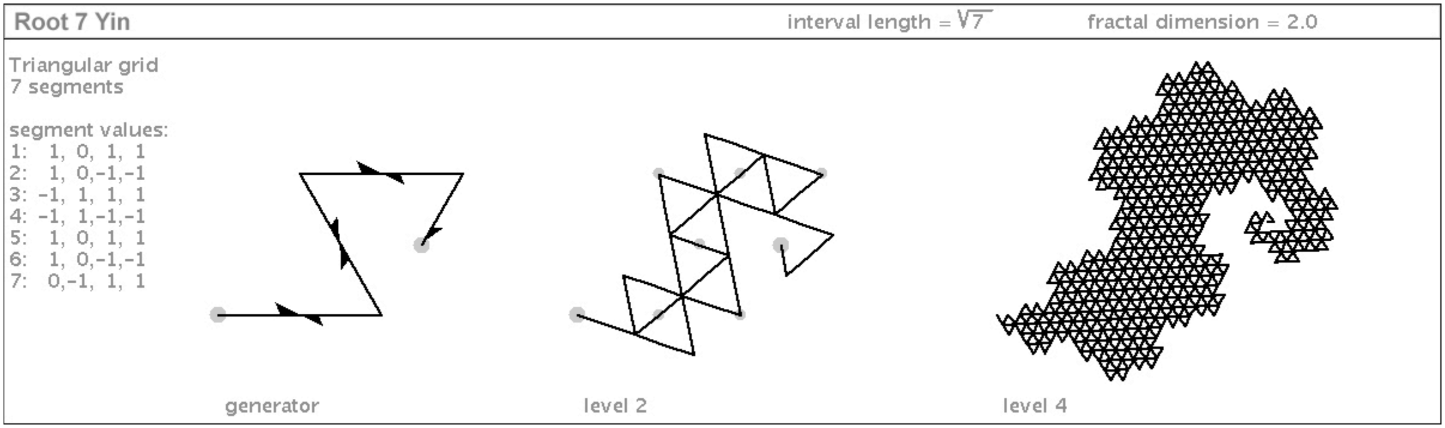


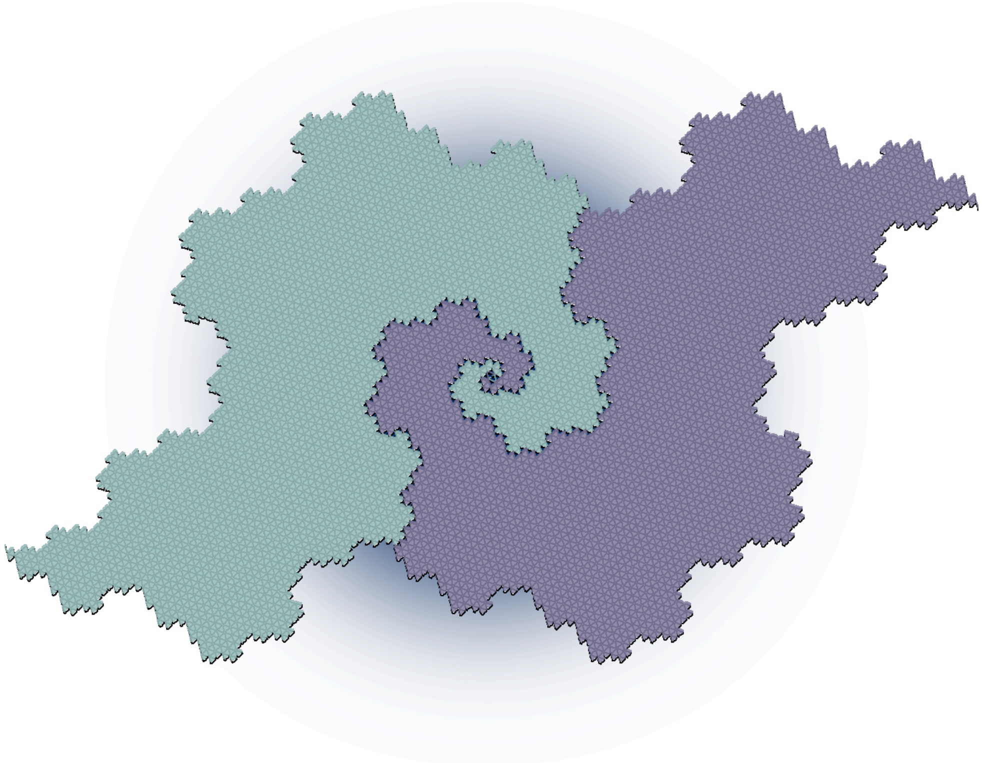


I have just shown you several variations of the 7-Dragon. Each generator visits the same seven grid points, but they are each visited in a different order. Below is a diagram of four ways that the seven points can be visited. It uses the rounded (chopped-off) corners to help with readability.

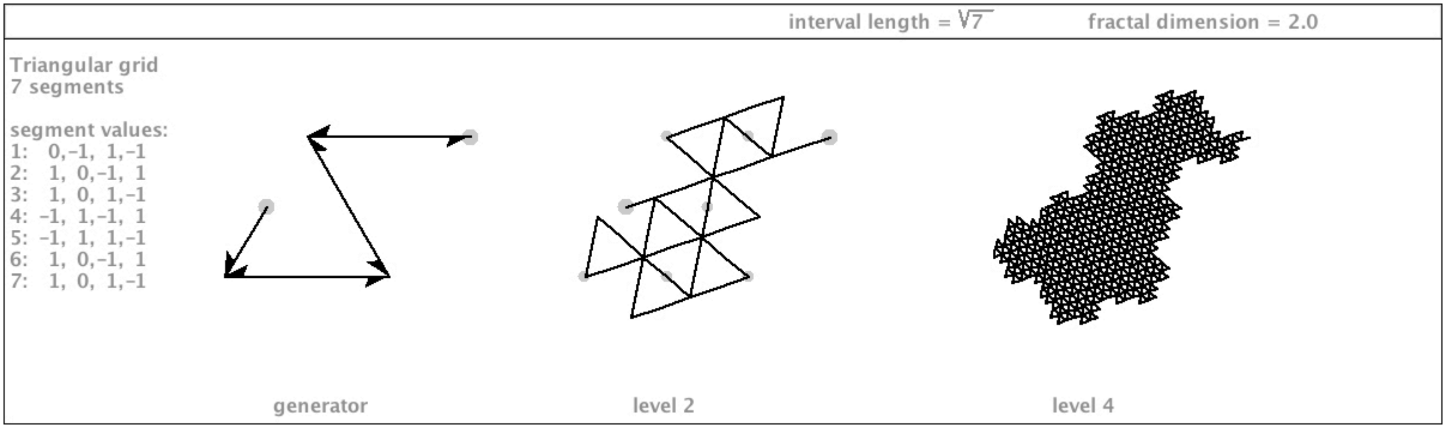


Here is a √7 gridfiller, with a distinctive hook. It looks a bit like the Yin Dragon I showed you earlier of the √3 family. Below is a rendering, pertiled with a 180-degree flipped partner. The two mate to create a yin-yang 7-dragon:

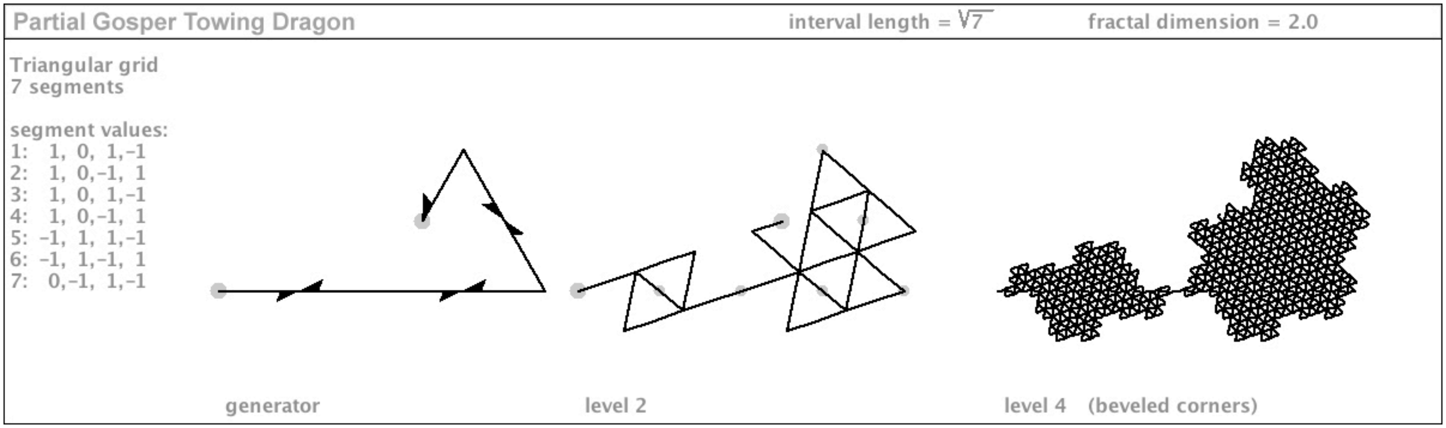




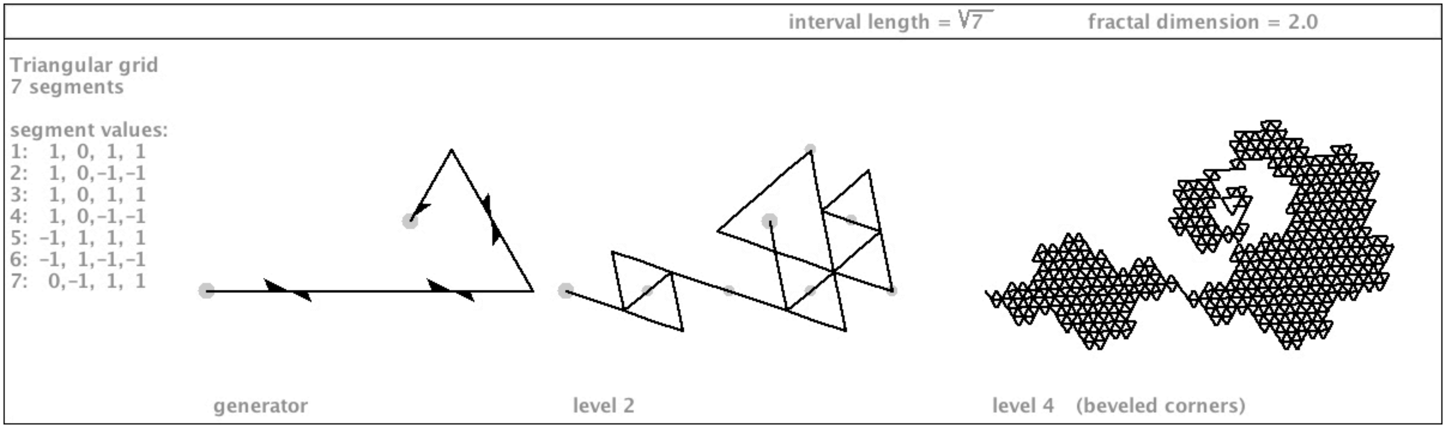
The profile of the 7-Dragon is similar to that of the Gosper Island – that is: the fat middle-section. As I mentioned before, the Gosper Island seems to pop up quite a bit in this family. Here is a curve that shows a bit of that profile (but only in certain parts).



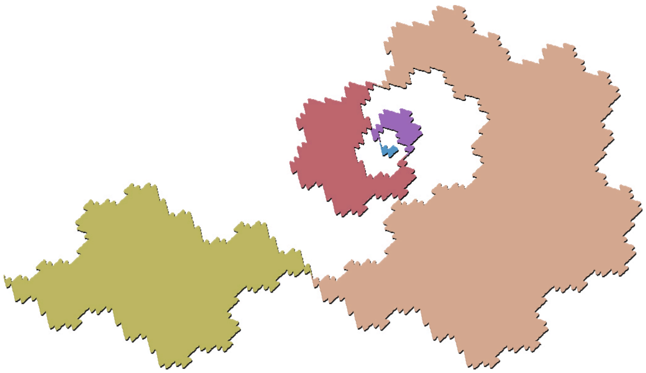
Here is a curious specimen. The main part of its body looks like a Gosper Island, but it is partly-eroded. And right below the eroded part is a 7-Dragon – tethered to its body.

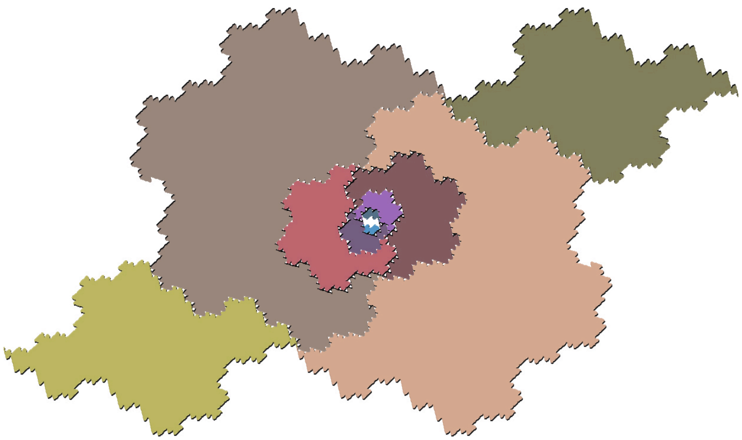


I discovered a variation of the specimen I just showed you. After careful analysis, I have concluded that this specimen had made a valiant attempt to fill up the missing piece of its Gosper body. But unfortunately, it had spent too many of its segments on its tethered 7-Dragon. And in its failed attempt to fill its Gosper body, it left a series of Gosper-like holes.

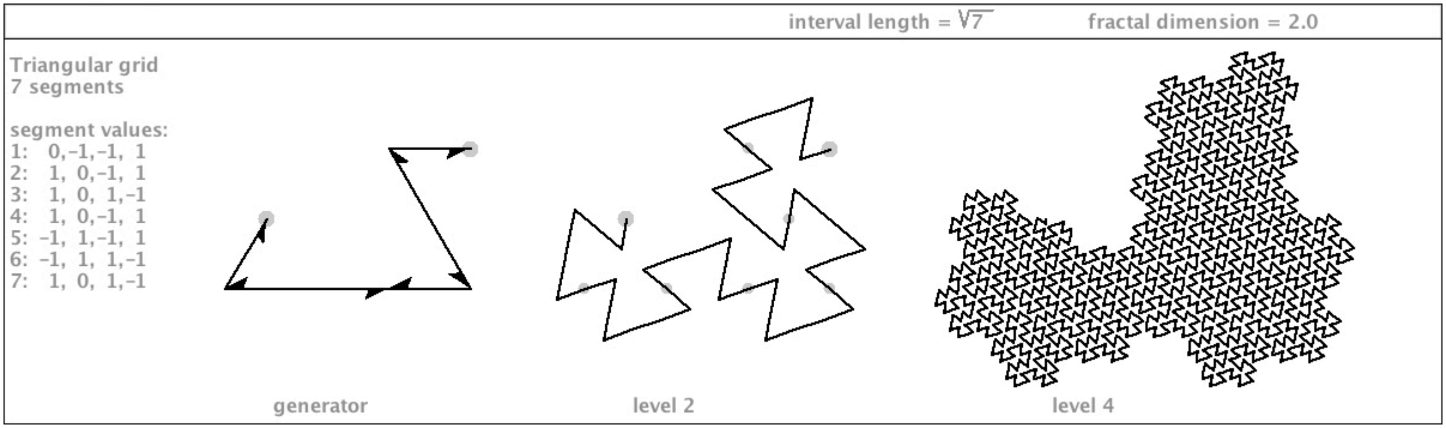


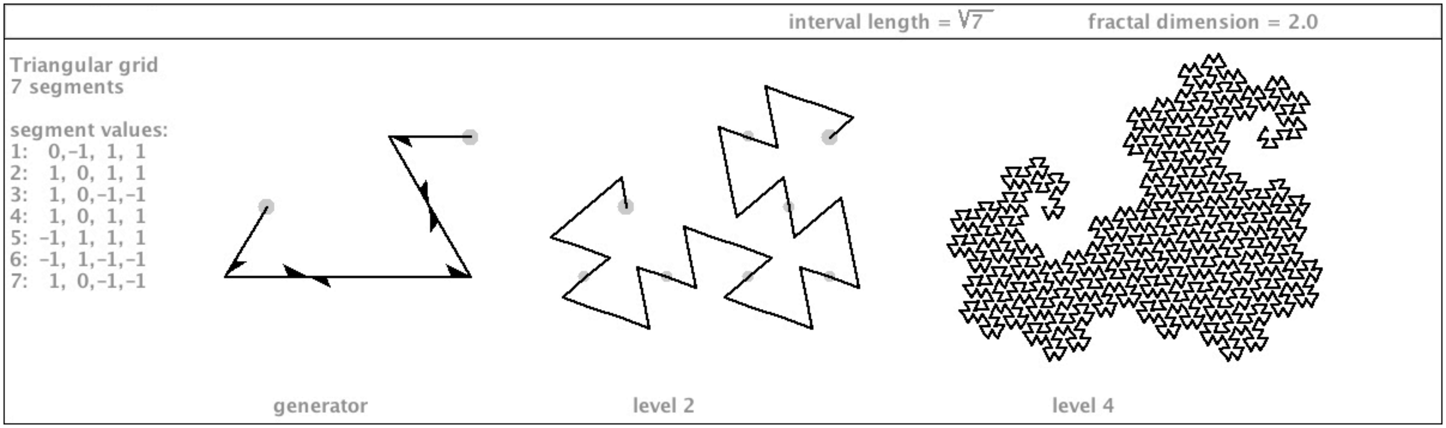
Below at left is a rendering of this specimen with regions colored to illustrate the Gosper pieces. Now, do you want to see what it looks like when two of these specimens mate? It is shown below at right. Don't try to wrap your whole brain around this one – you might get a headache.

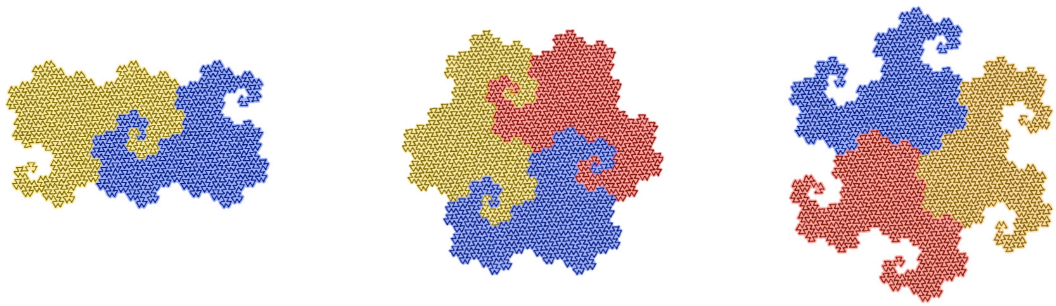




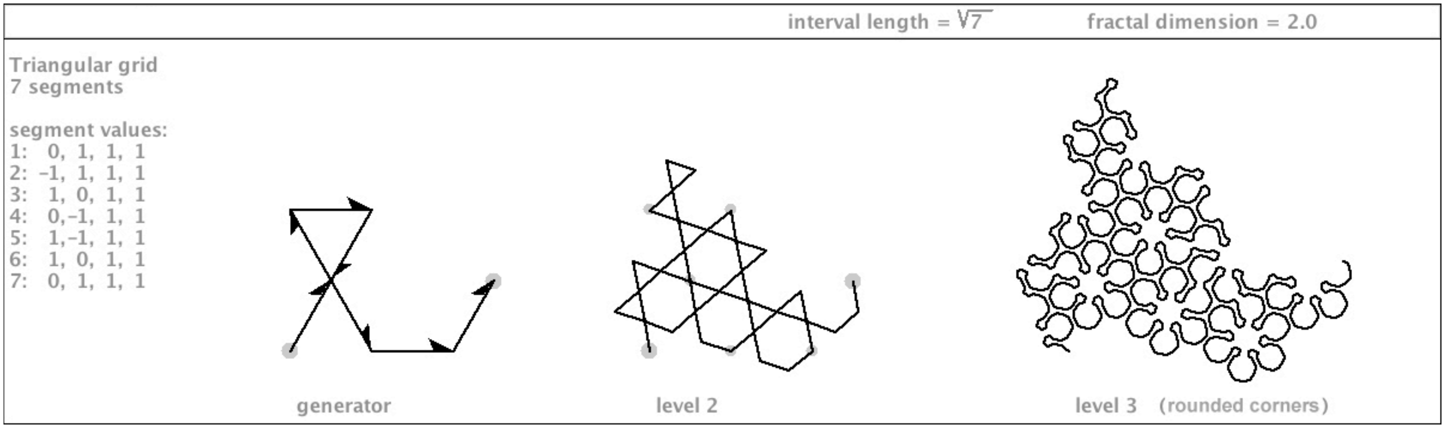
The Gosper profile shows up again in these two variations on one generator. They both have a rather jaggy inner-texture. You can see in the level 2 teragon below that these shapes are still quite similar. But at level four, it becomes clear that the second specimen has two spiraling hooks. This specimen is shown below with a few rep-tiling themes.

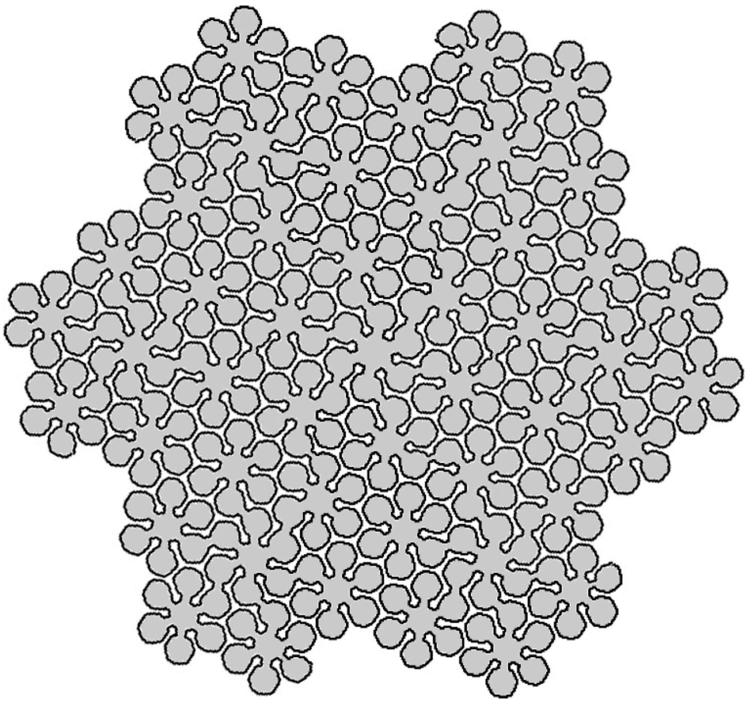




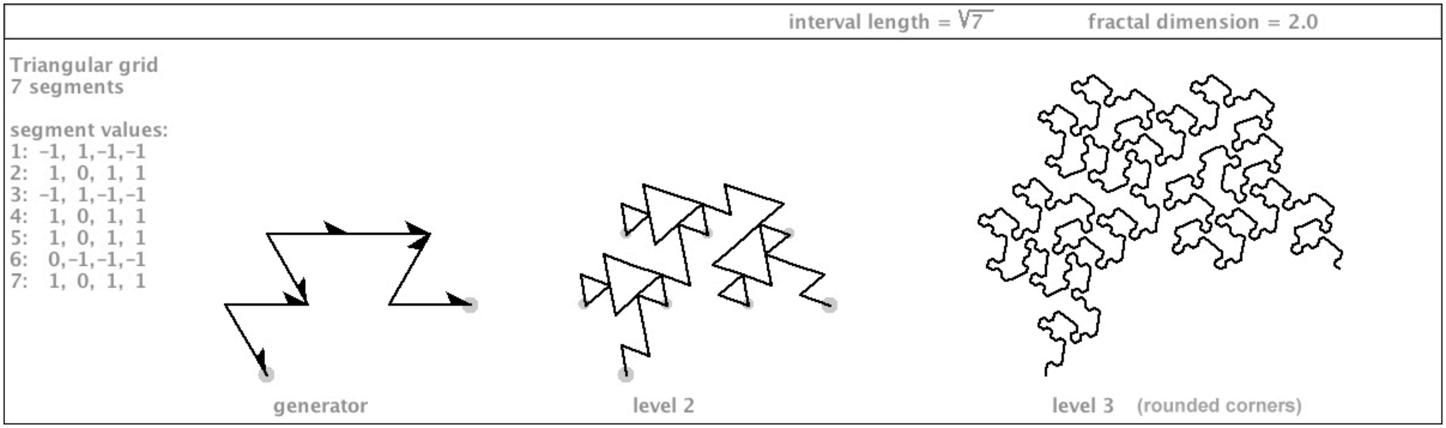


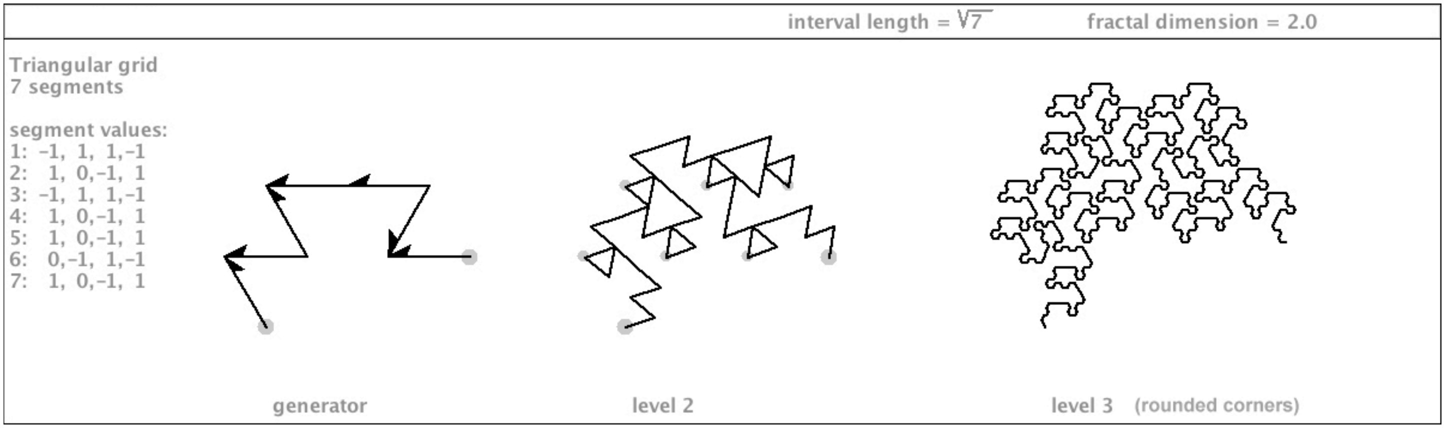
Gosper profile cleverness abounds: This specimen is pertiled 6 times to form a closed Gosper Island.

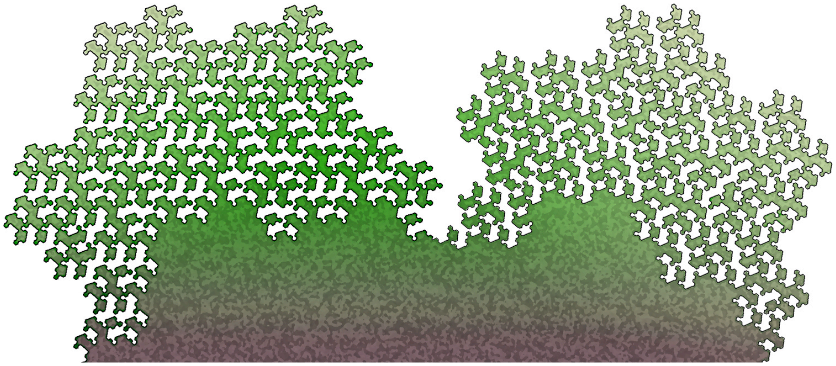




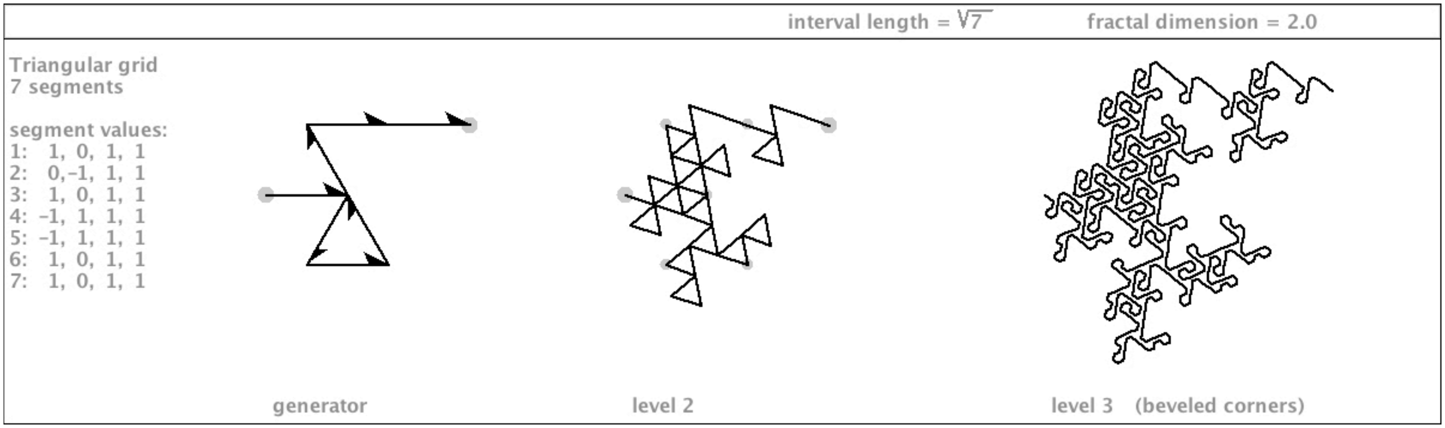
The two specimens below create variations on a “Gosper Hill”, covered with lush vegetation, shown below:

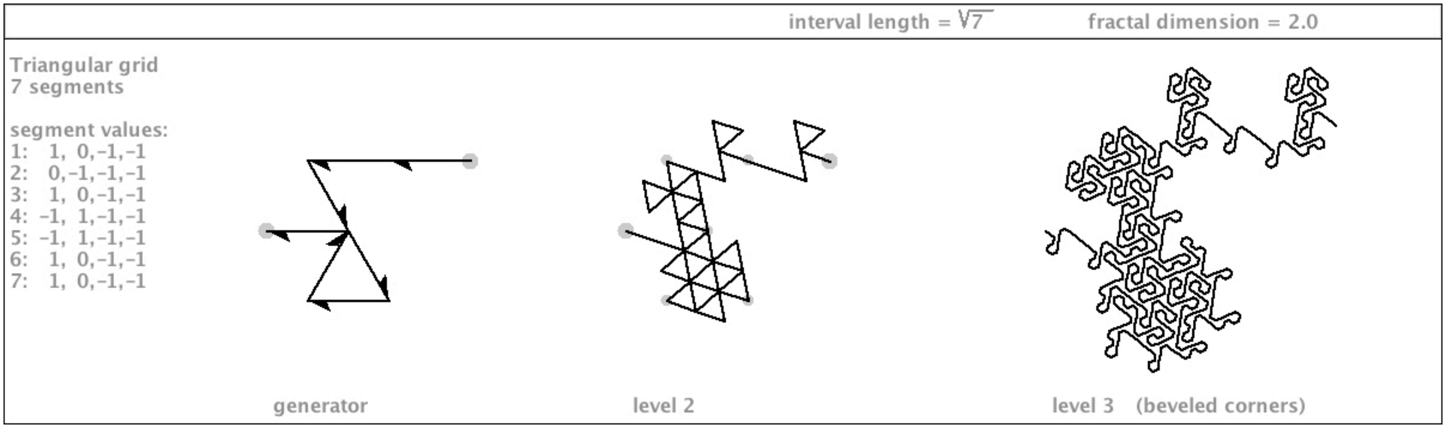


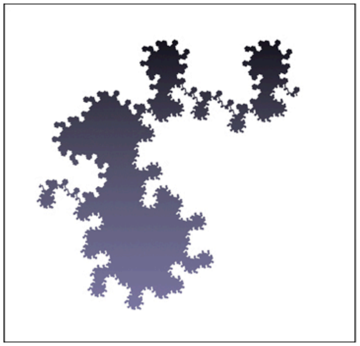


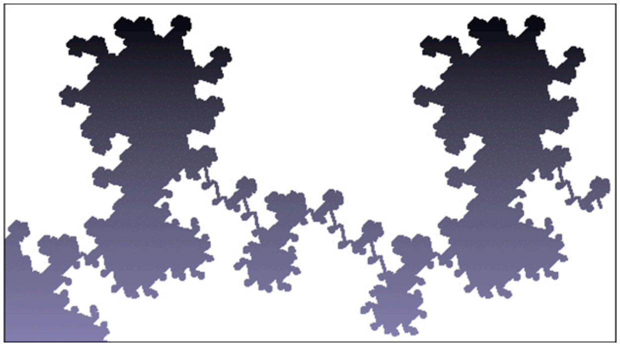


Here are two flip-variations of a generator. They are gridfillers. The second one is shown below with an enlarged area.

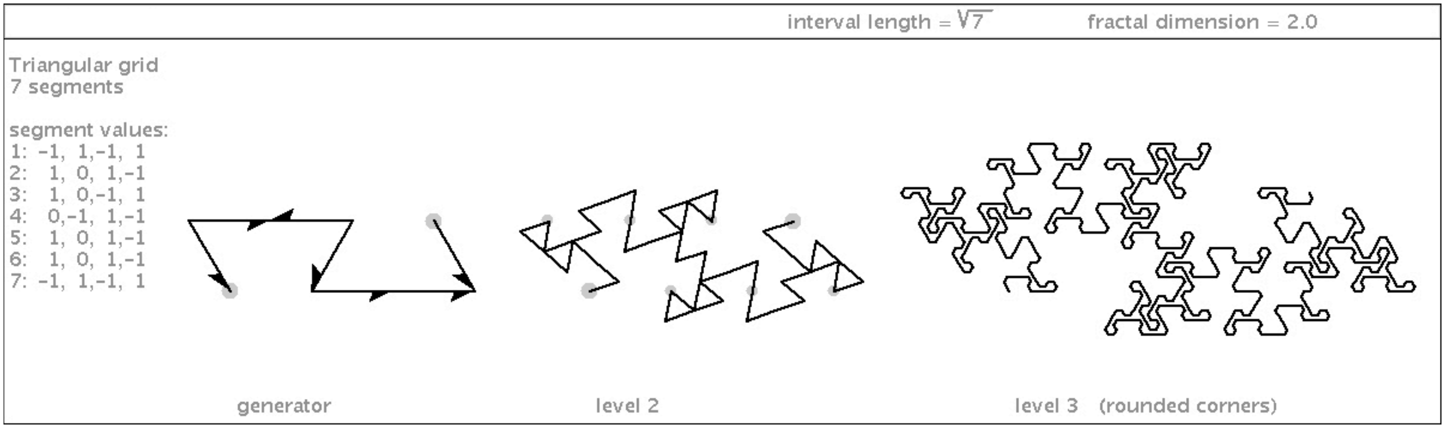


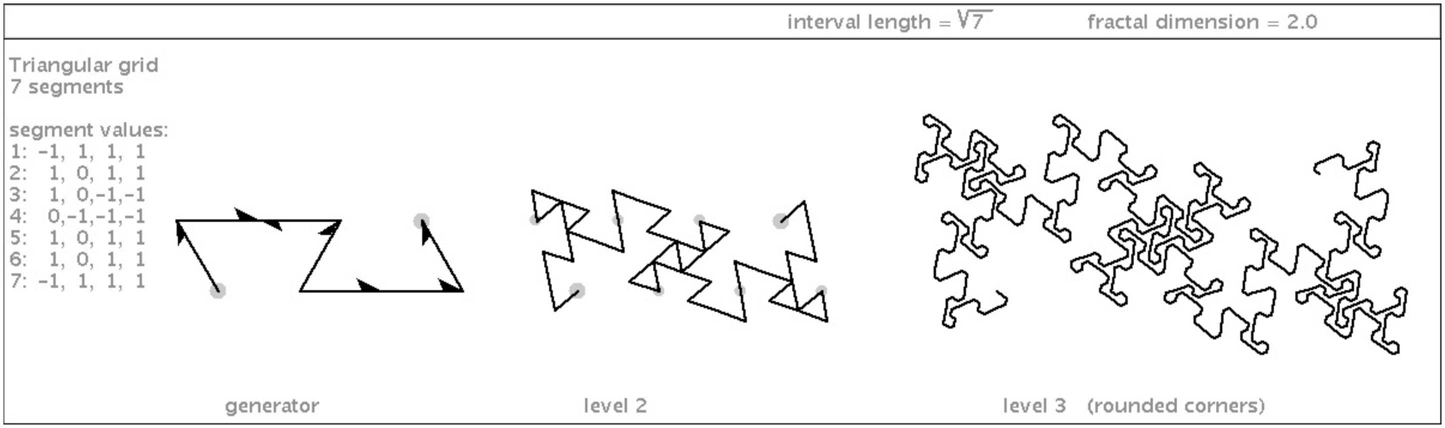


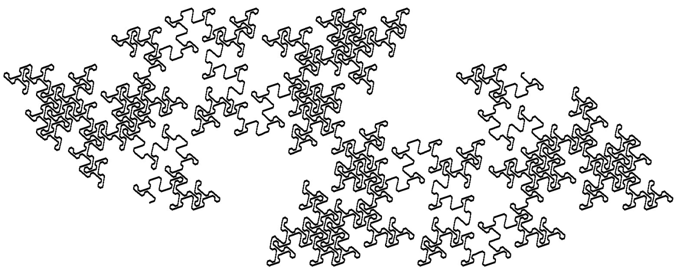


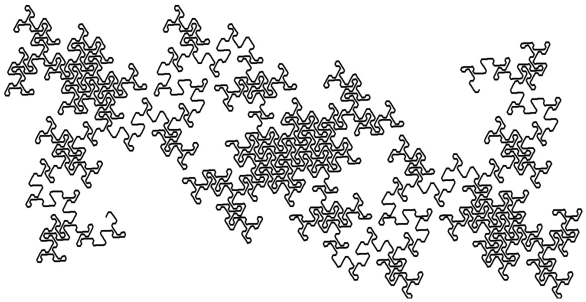


Here are two specimens based on a common generator shape.

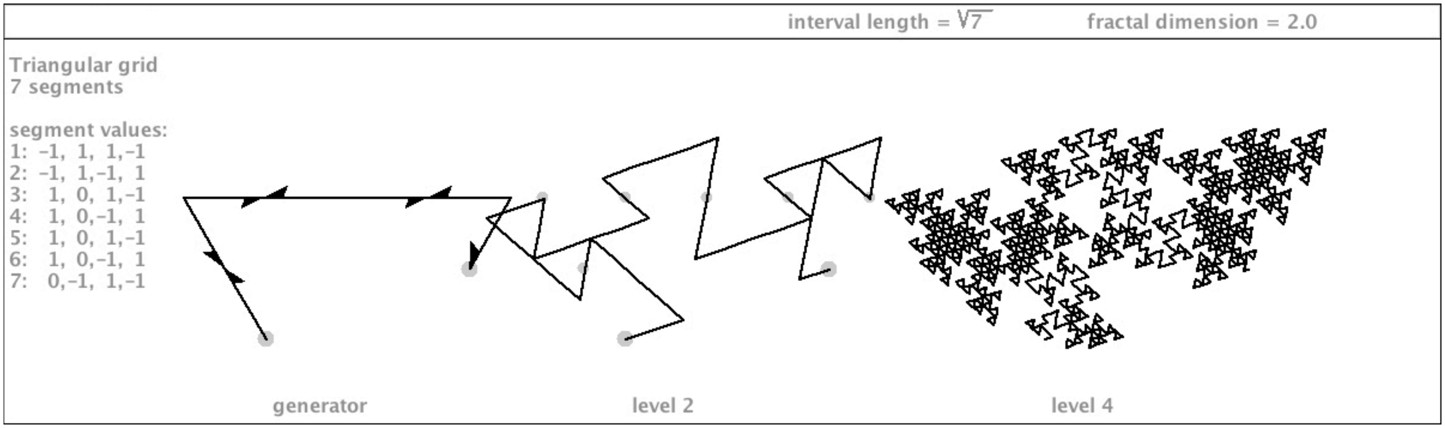


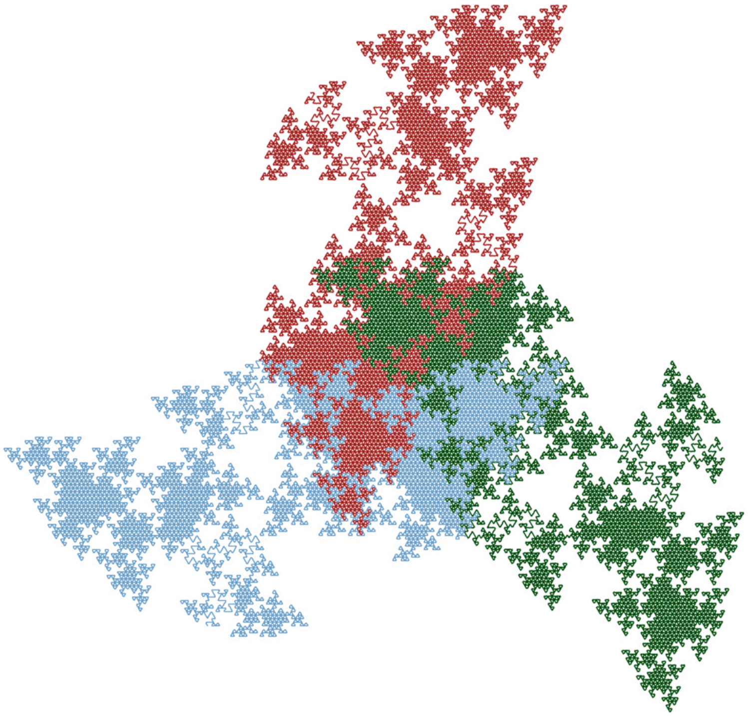




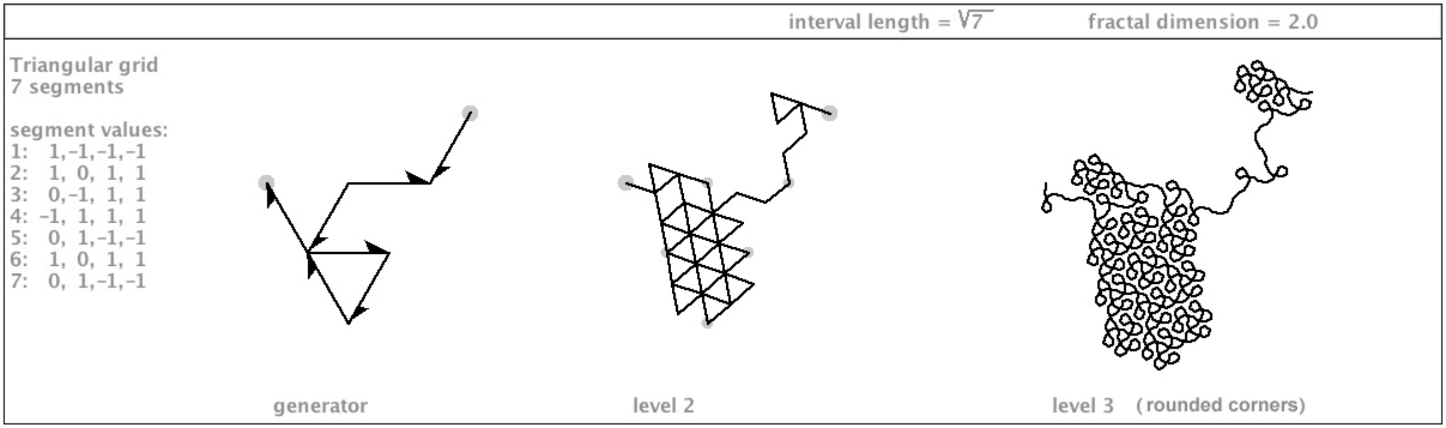


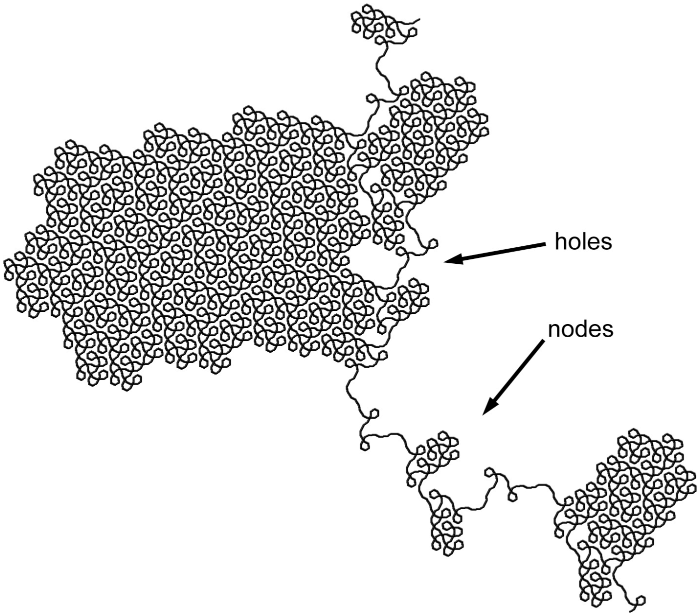
Here is a highly craggy specimen. It is shown below in a 3-way pertiling.





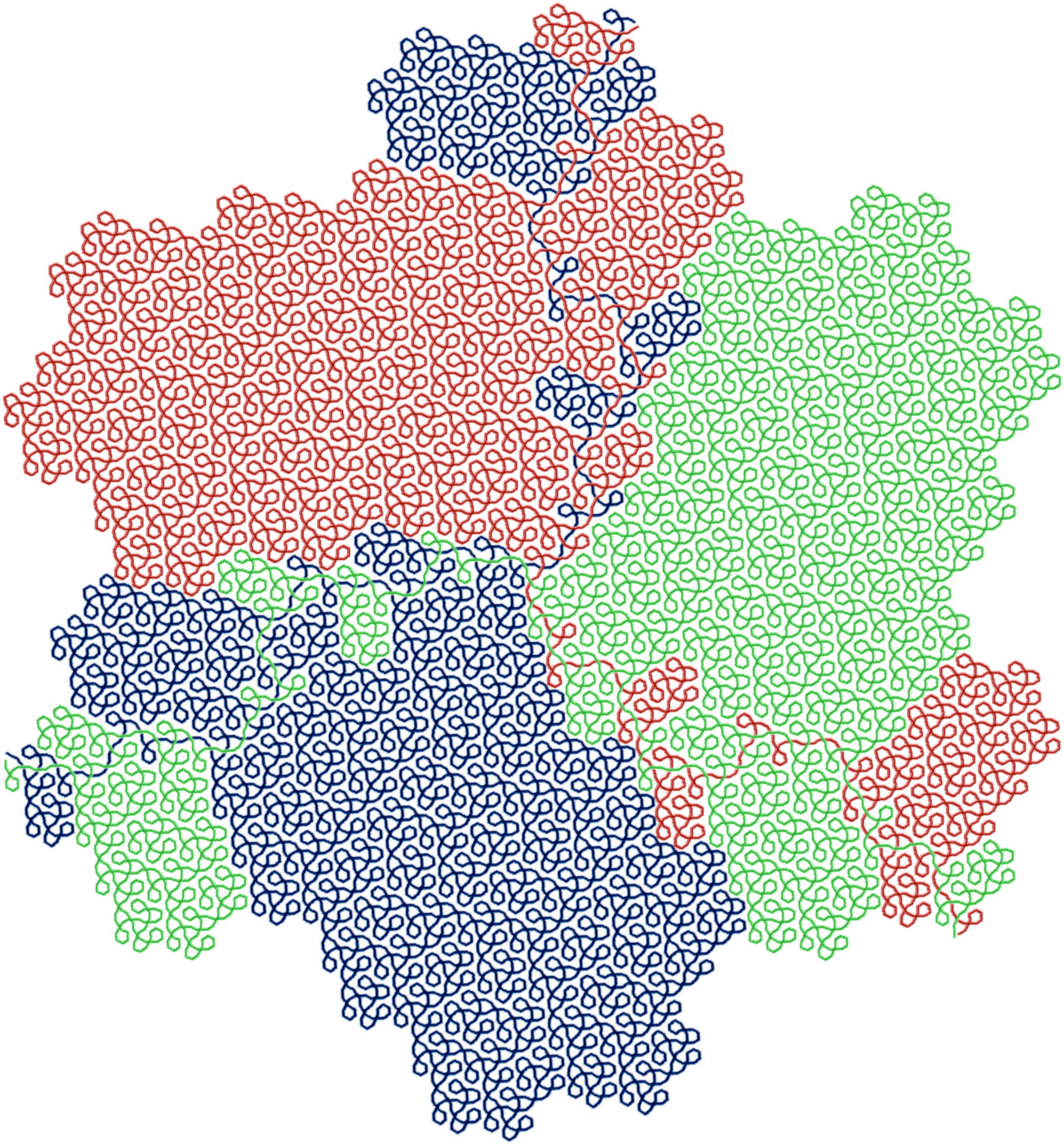
This self-crossing curve is quite interesting. Below is a rotated rendering of teragon 4.



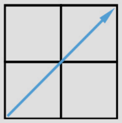


On the next page I show something about this curve that I had not initially expected. This may not come as such a surprise to you, after having seen several specimens. When I first drew this curve, I noticed that the *holes* in the shape corresponded to the *nodes*. Could it be that the nodes could fit into the holes?

So I decided to try some pertiling: I made a copy and rotated it by 120 degrees (knowing that this is a triangle-grid specimen). And lo and behold, they fit together like peas in a pod (or several peas in several pods). Then I wondered if the remaining holes might be filled by a third copy, rotated by -120 degrees. Imagine how excited I was when the three of these specimens fit together to form a Gosper Island! On the next page is a picture of this three-way mating. That's quite an intimate embrace!

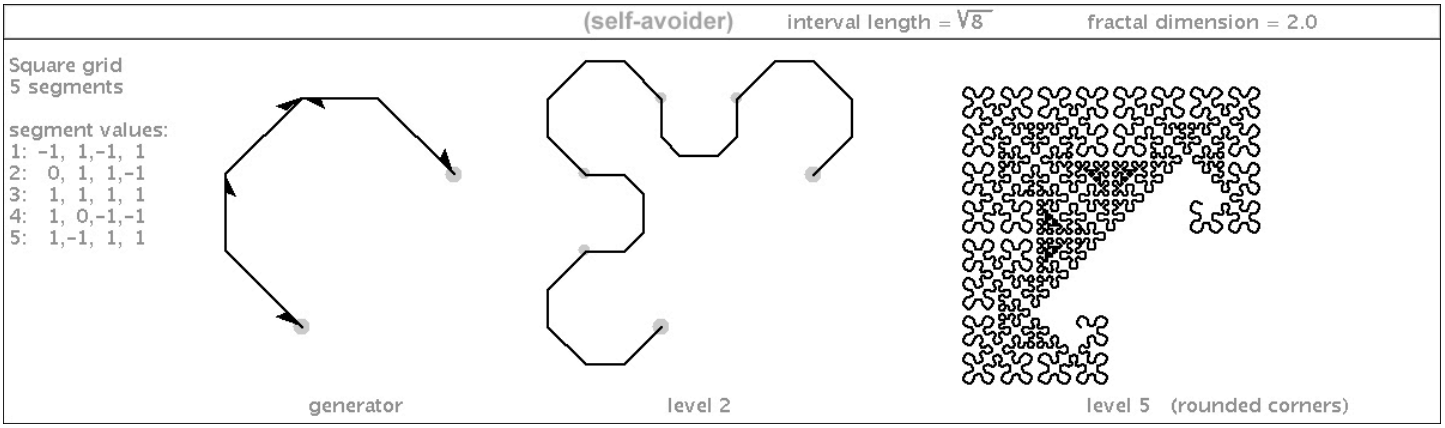


√8



Next we come to the √8 family. This family has some things in common with the √2 family: the generator lies on a 45 degree diagonal. Also, it can be seen as a *superset* of the √2 family (as well as the √4 square grid family). This is because the values 2, 4, and 8 are powers-of-two numbers ($2^1 = 2$; $2^2 = 4$; $2^3 = 8$). We will see later that the √16 square grid family is a superset of the √2, √4 square grid, and √8 families.

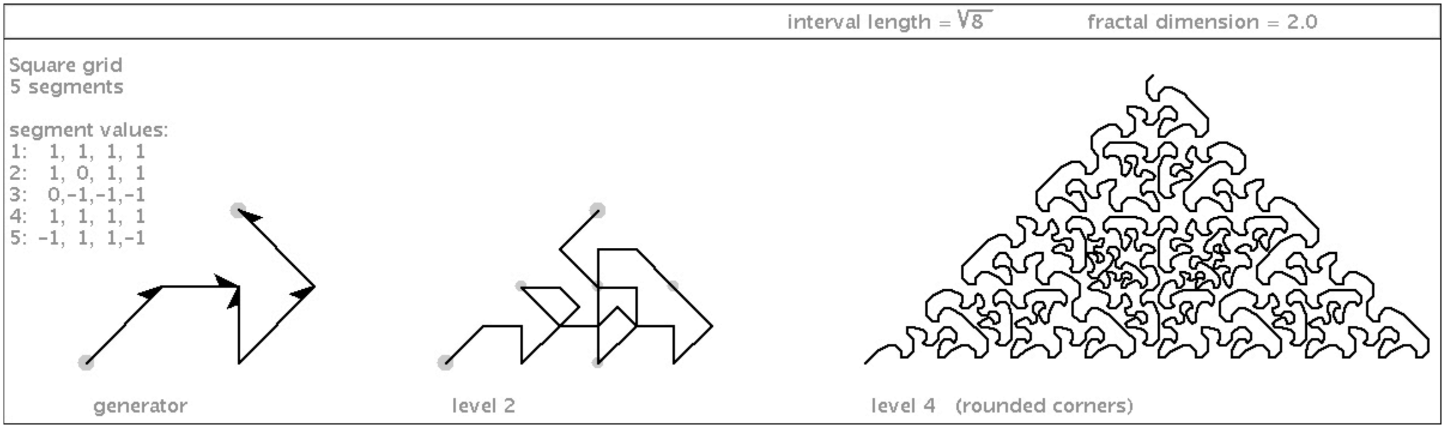
We've already seen one member of the √8 family: I showed it to you at the beginning of the book as an example of how the turtle can use flippings to convert an otherwise self-crossing curve into a self-avoiding curve:

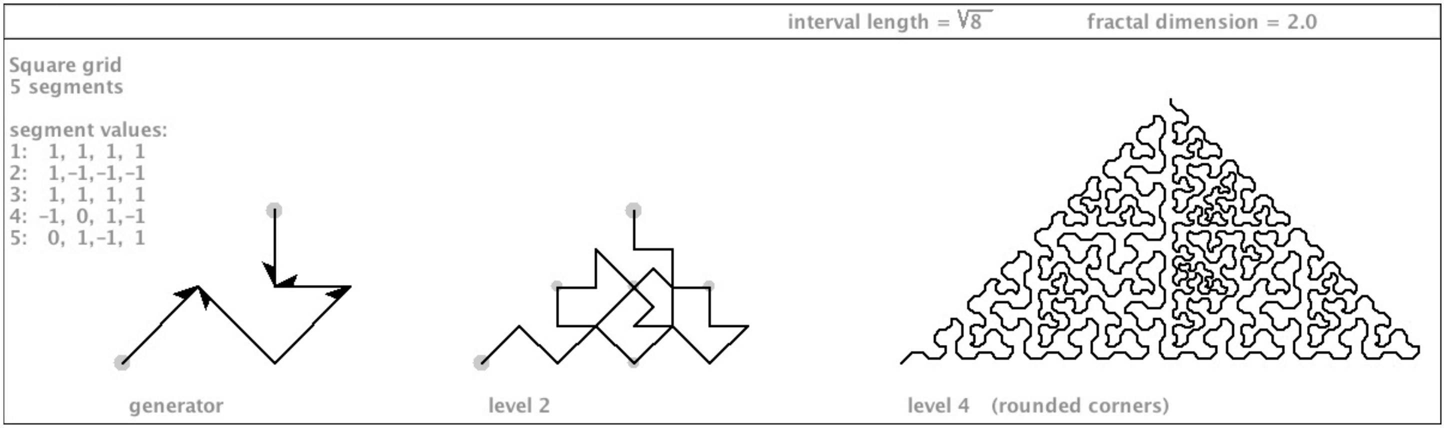


When I showed this curve to you before, it was rotated 45 degrees. Here is it shown in its native familial orientation. Notice that the generator has only 5 segments, and that three of those segments have a length of √2. These segments are responsible for the three large lobes in the 2^{nd} teragon.

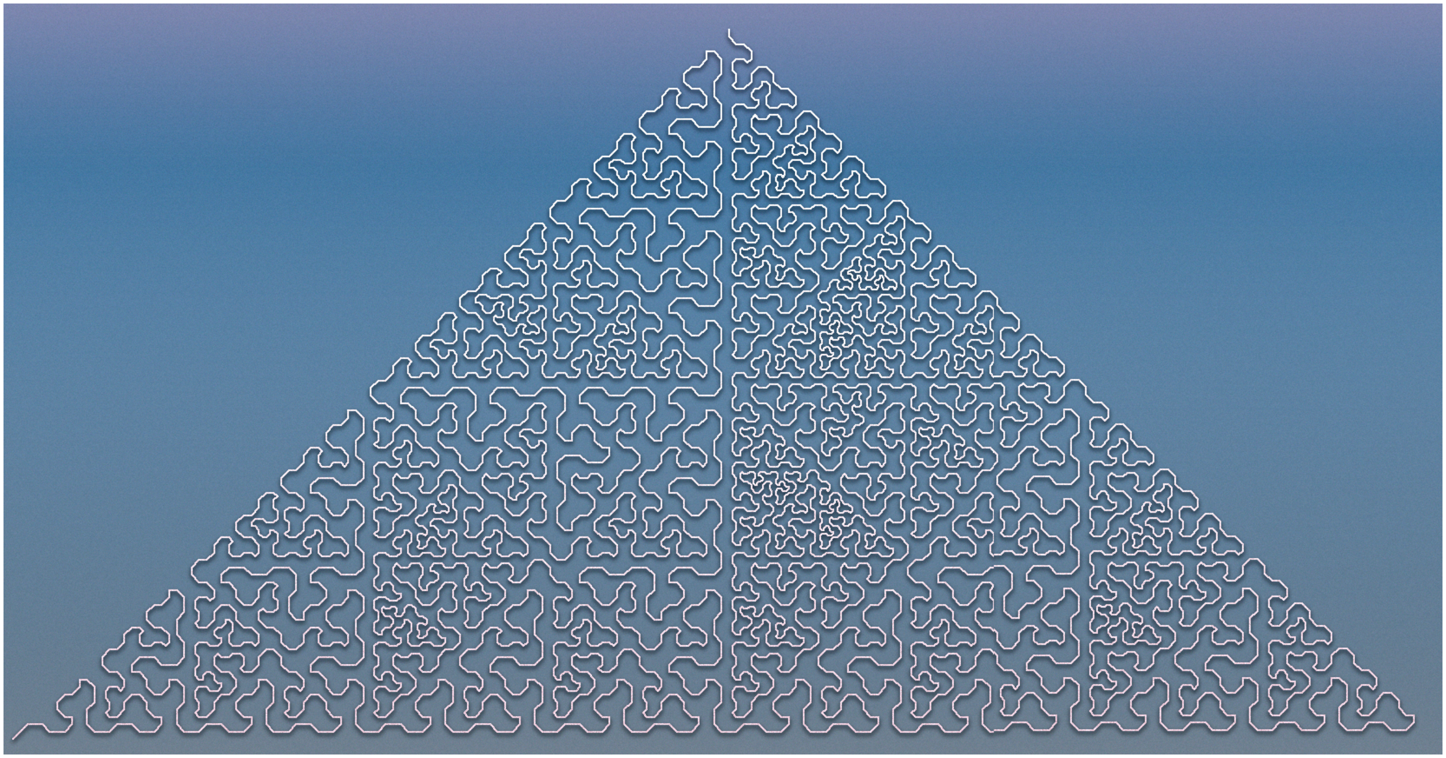
As a general rule, you can consider segments of length √2 to count as *two* (remember that we square the lengths when calculating fractal dimension). So in this case, two one-length segments and three segments of √2 – when squared – add up to 8: the family number.

The √8 family is quite versatile. First let’s look at some curves that fill a right triangle. Two of them are shown below. Like the last curve I showed you, the generators for these curves each have 5 segments, three with a length of √2.

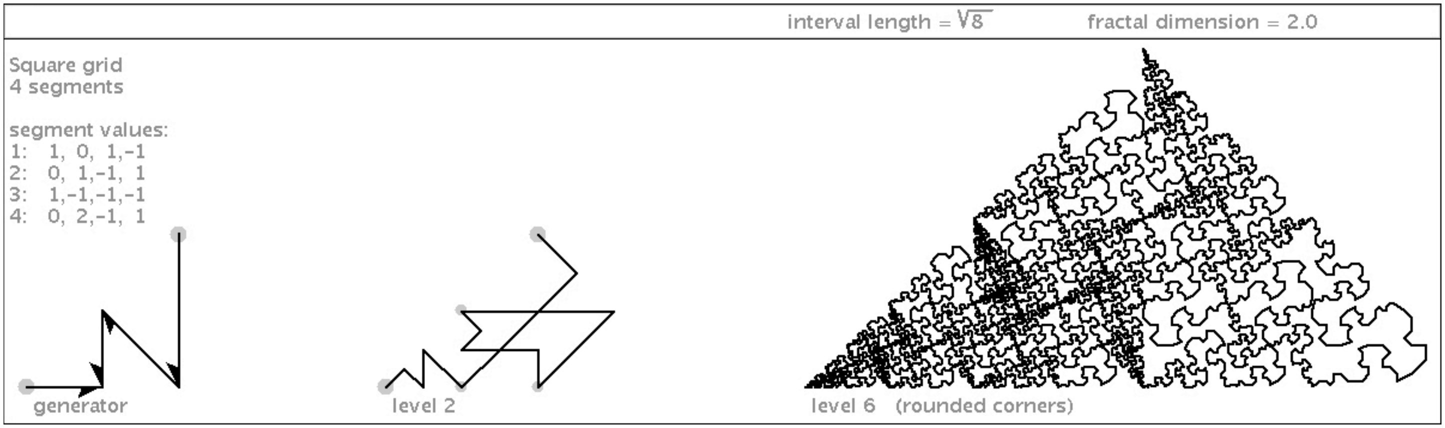




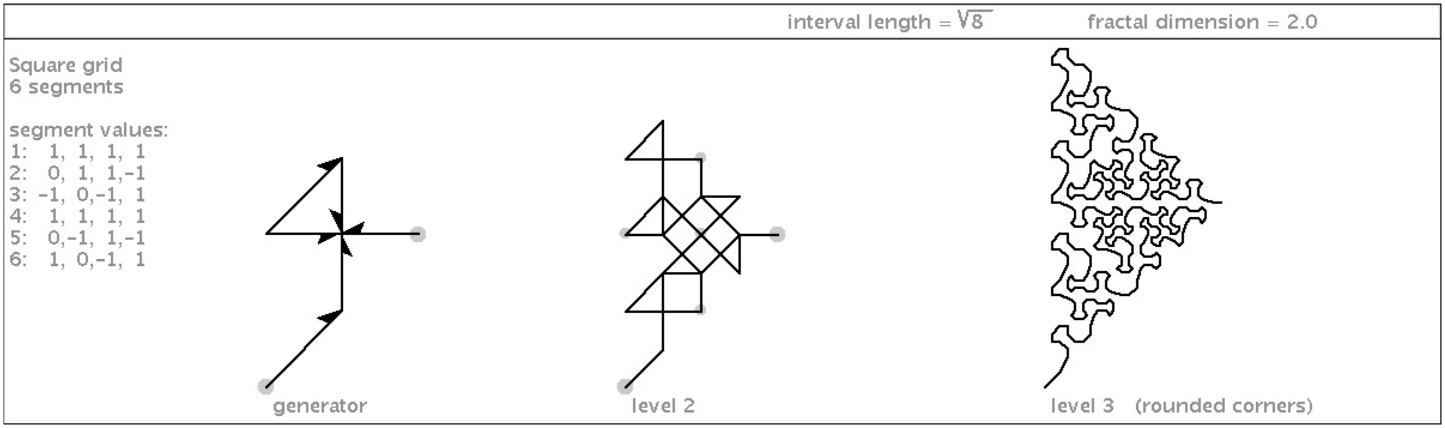
This last one is so interesting I decided to render it in color. It’s shown on the next page.

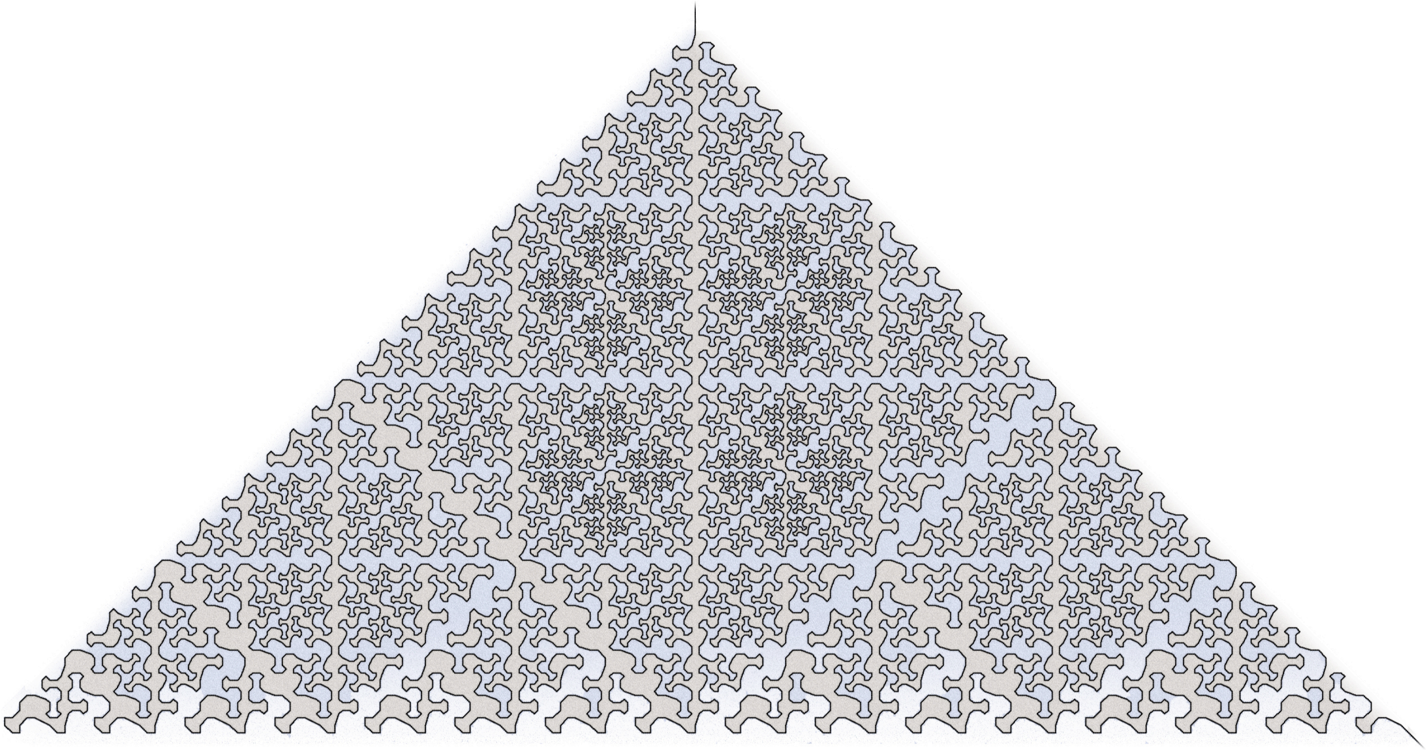


Here's another triangle-shaped specimen. This one has even fewer segments: 4! One of the segments is length √2 and the last segment is length 2. Following the rule of squaring all lengths, as I said earlier, you can see how the sum is 8. Because of the long segment length, the result has a great variety of lengths within – with a lot of self-similar patterning.

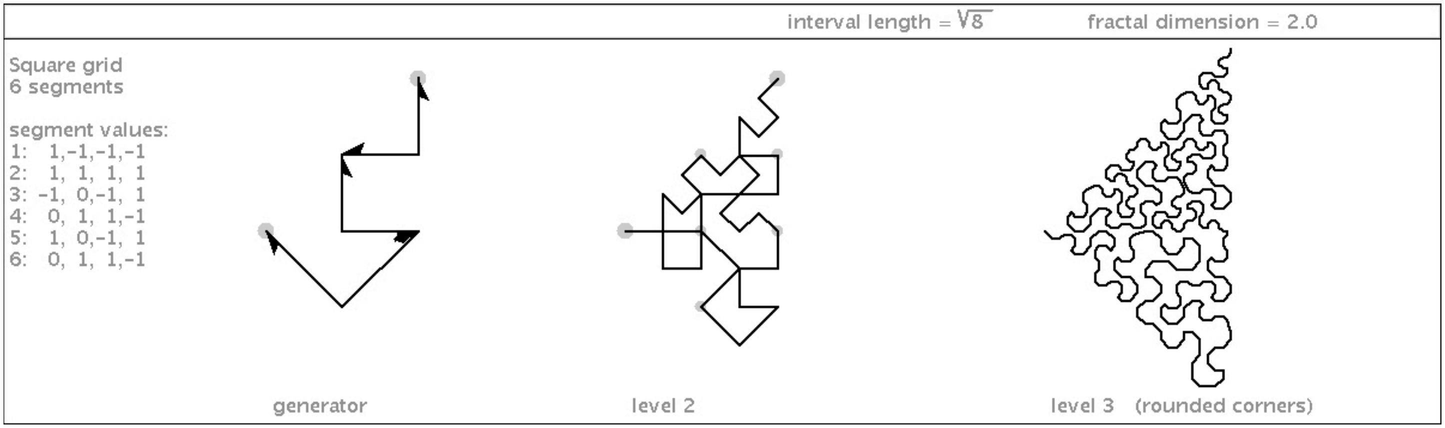


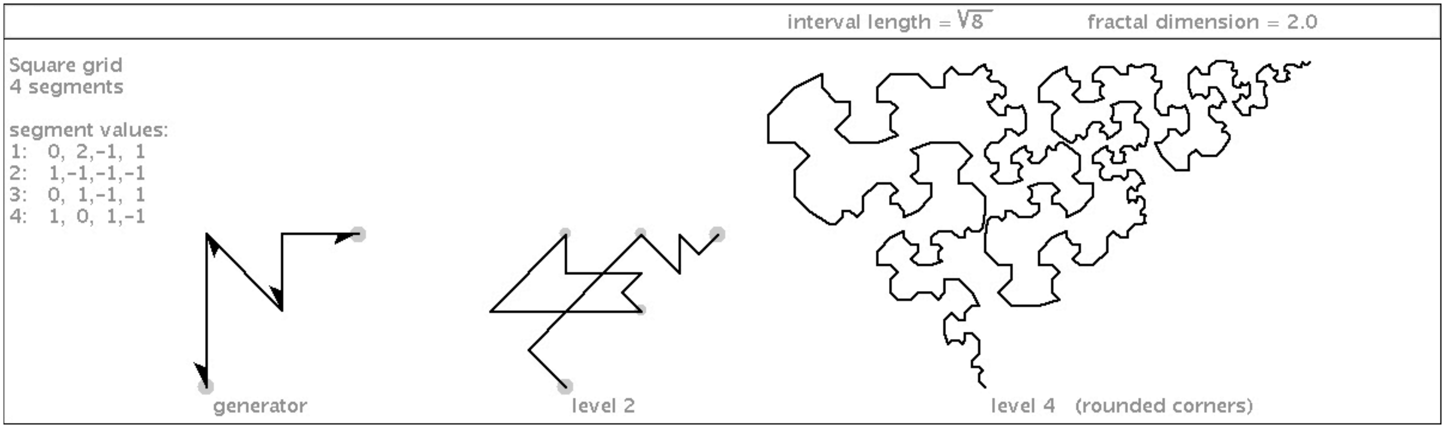
I’m particularly fond of this one. So I made a color rendering of it, and rotated it 90 degrees. It is shown below.

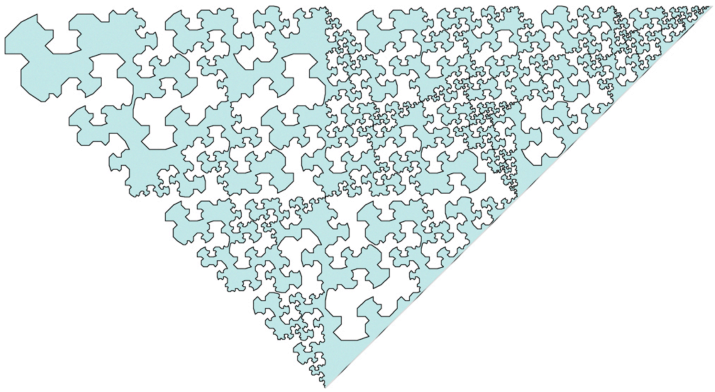




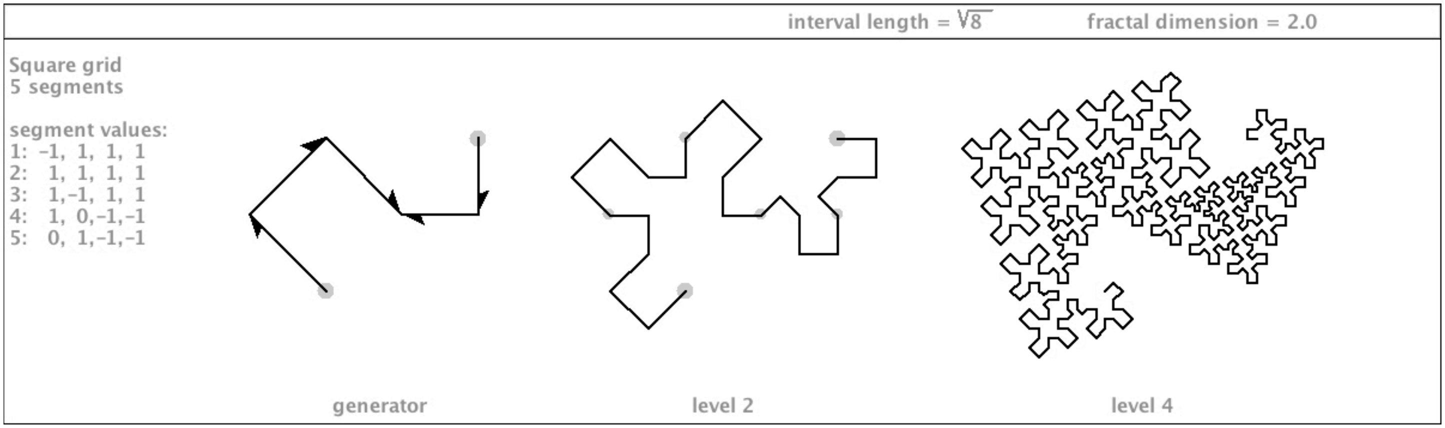
Here are two more right triangles of the √8 family. These are a little less well-behaved, but interesting nonetheless.

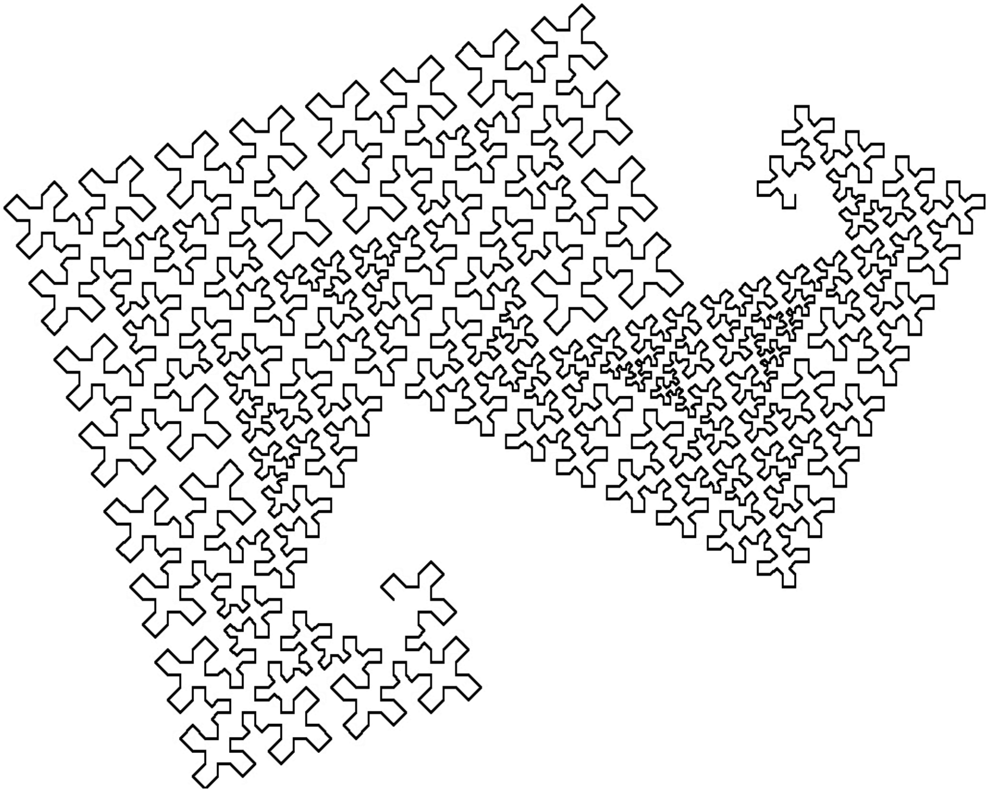


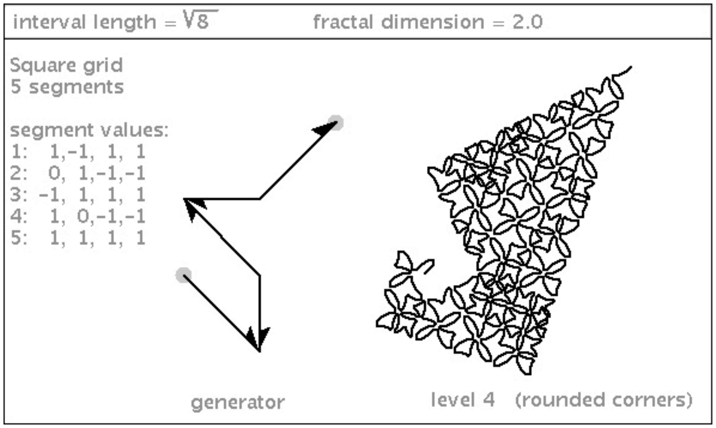


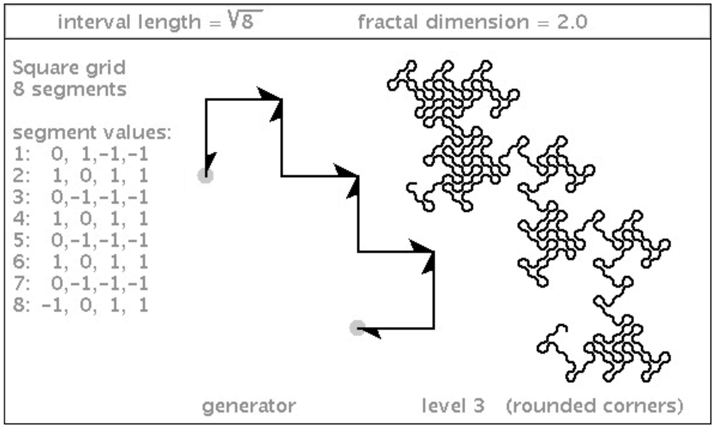


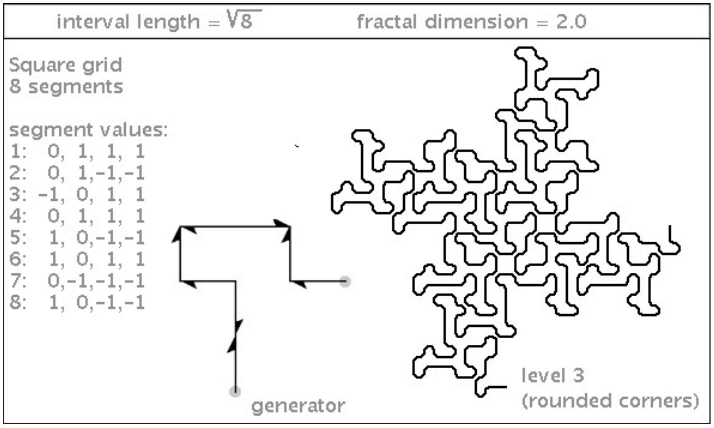
This next specimen is a natural self-avoider. It is a relative of a √4 specimen we met earlier. On the next page I show five more curves of this family.

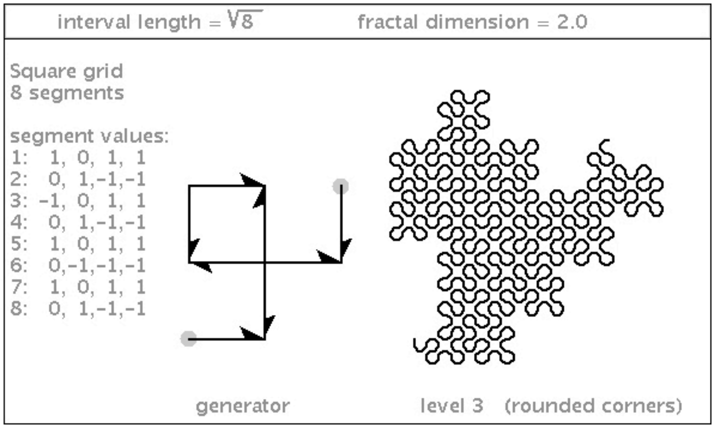


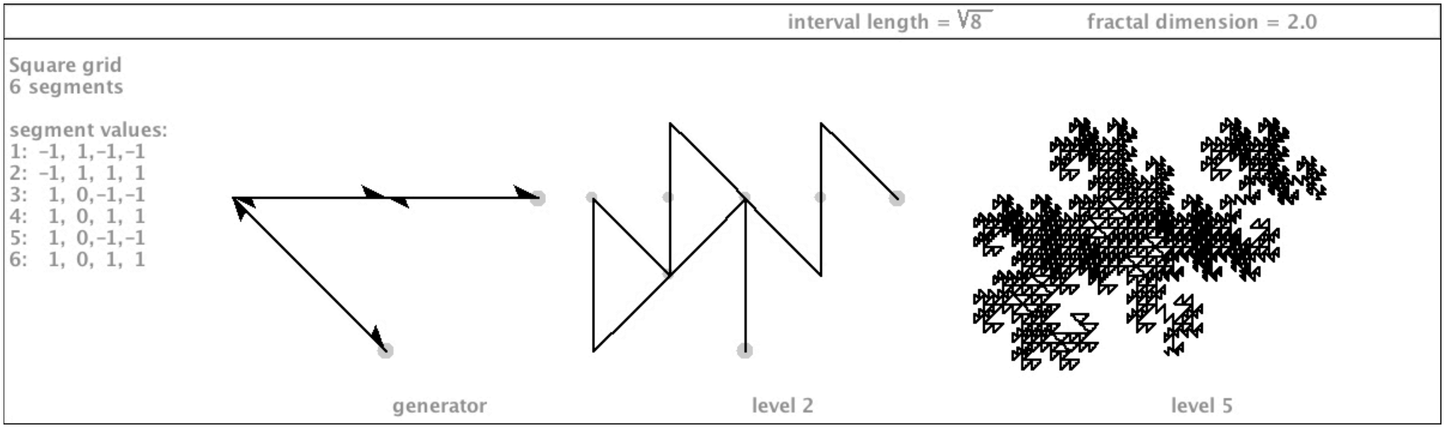



interval length = √8
fractal dimension = 2.0
Square grid
5 segments
segment values:
1: 1,-1, 1, 1
2: 0, 1,-1,-1
3: -1, 1, 1, 1
4: 1, 0,-1,-1
5: 1, 1, 1, 1
generator
level 4 (rounded corners)

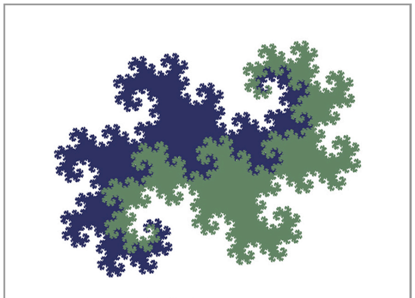

interval length = √8
fractal dimension = 2.0
Square grid
8 segments
segment values:
1: 0, 1,-1,-1
2: 1, 0, 1, 1
3: 0,-1,-1,-1
4: 1, 0, 1, 1
5: 0,-1,-1,-1
6: 1, 0, 1, 1
7: 0,-1,-1,-1
8: -1, 0, 1, 1
generator
level 3 (rounded corners)

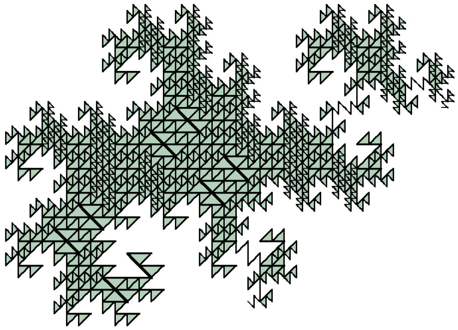

interval length = √8
fractal dimension = 2.0
Square grid
8 segments
segment values:
1: 0, 1, 1, 1
2: 0, 1,-1,-1
3: -1, 0, 1, 1
4: 0, 1, 1, 1
5: 1, 0,-1,-1
6: 1, 0, 1, 1
7: 0,-1,-1,-1
8: 1, 0,-1,-1
generator
level 3
(rounded corners)


interval length = √8
fractal dimension = 2.0
Square grid
8 segments
segment values:
1: 1, 0, 1, 1
2: 0, 1,-1,-1
3: -1, 0, 1, 1
4: 0, 1,-1,-1
5: 1, 0, 1, 1
6: 0,-1,-1,-1
7: 1, 0, 1, 1
8: 0, 1,-1,-1
generator
level 3 (rounded corners)

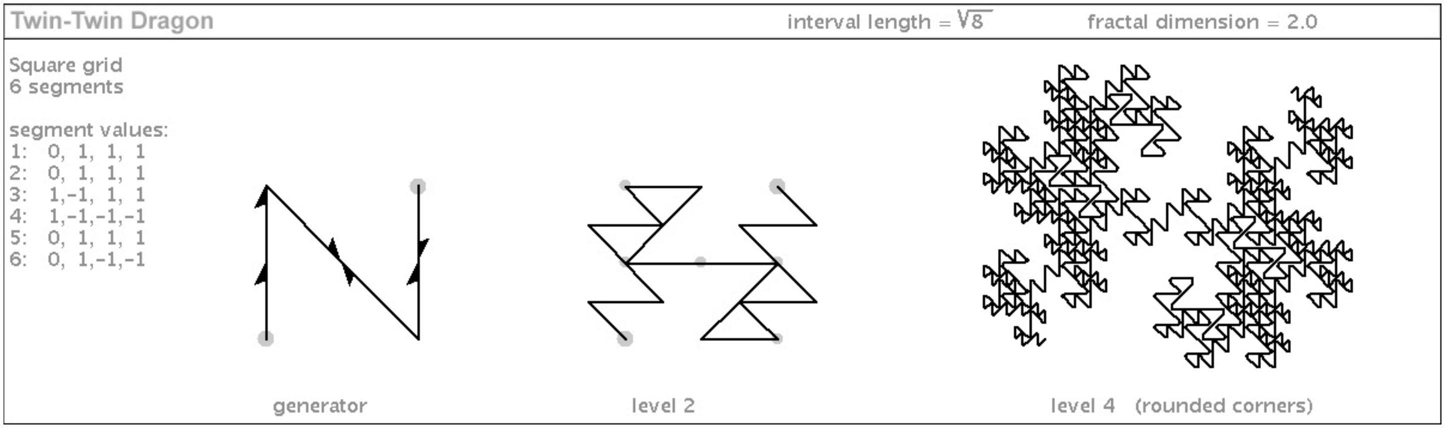

interval length = √8
fractal dimension = 2.0
Square grid
6 segments
segment values:
1: -1, 1,-1,-1
2: -1, 1, 1, 1
3: 1, 0,-1,-1
4: 1, 0, 1, 1
5: 1, 0,-1,-1
6: 1, 0, 1, 1
generator
level 2
level 5

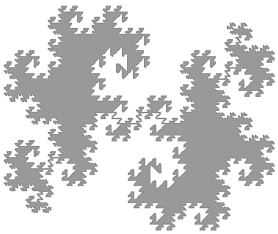
That last curve fractalizes into a shape that is similar to the “Twin Dragon”: the result of joining two HH Dragons. But notice that it is not quite the same as the Twin Dragon; it has pinched-off babies – which each have their own pinched-off babies.

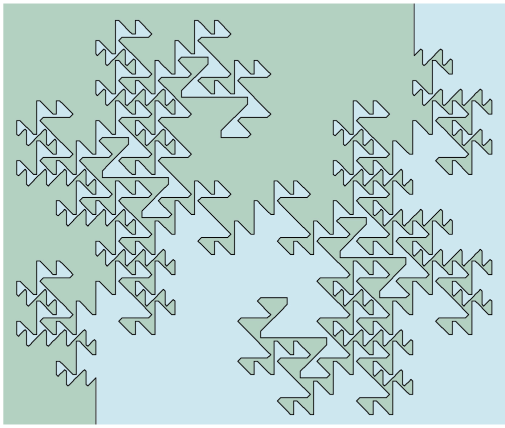

Twin Dragon



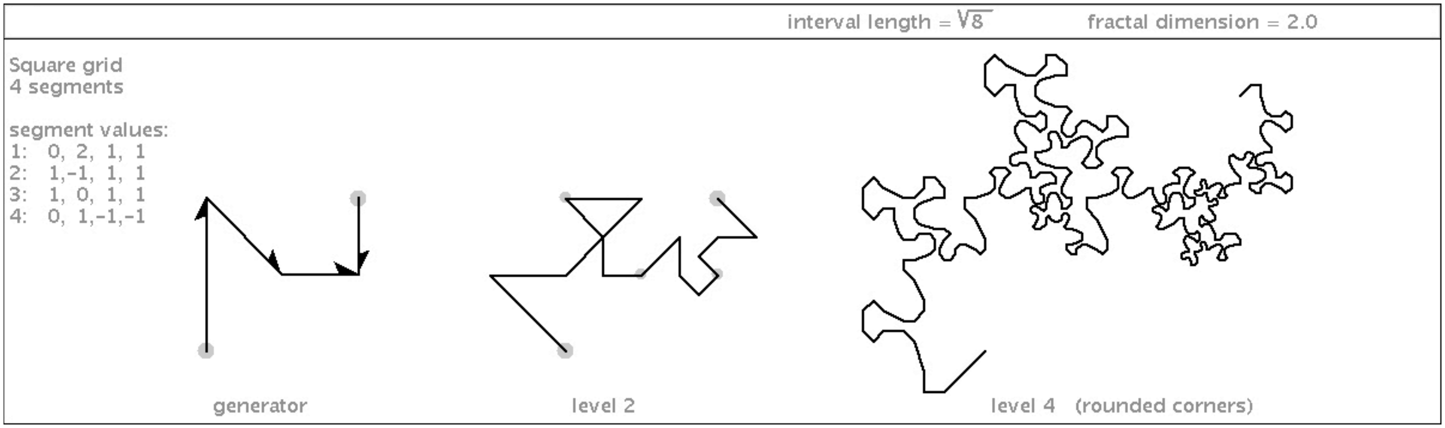
This next curve fractalizes into a *pair* of Twin-Dragon-like curves. Because of the similarity to the shape as the Twin Dragon, I call it the “Twin-Twin Dragon”. Are the two twins holding hands? No; their babies are holding baby hands.



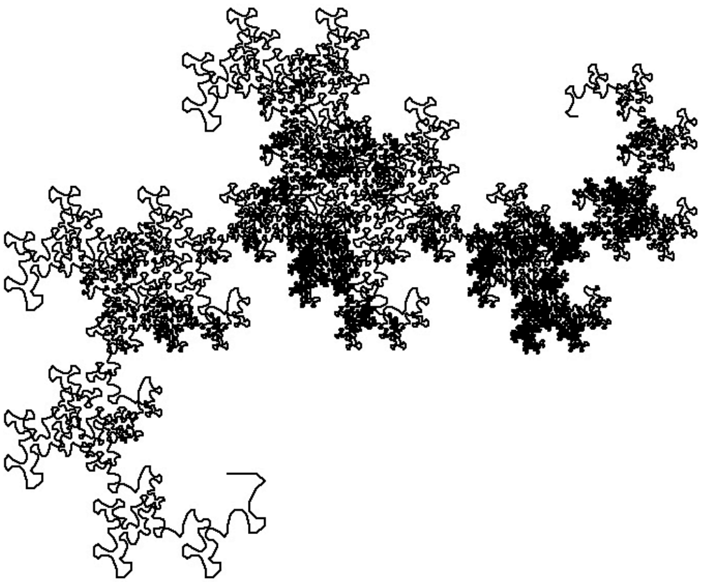

Twin-Twin Dragon

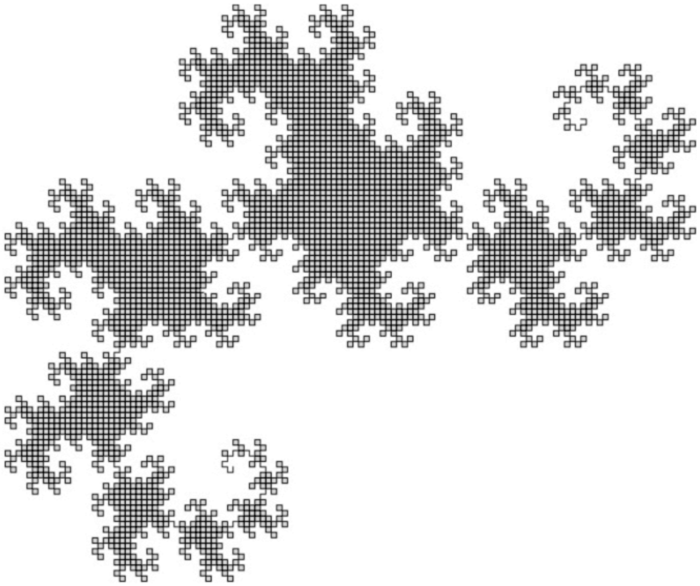


Speaking of dragons, the √8 family produces more dragons that are related to the HH Dragon. Here's one:

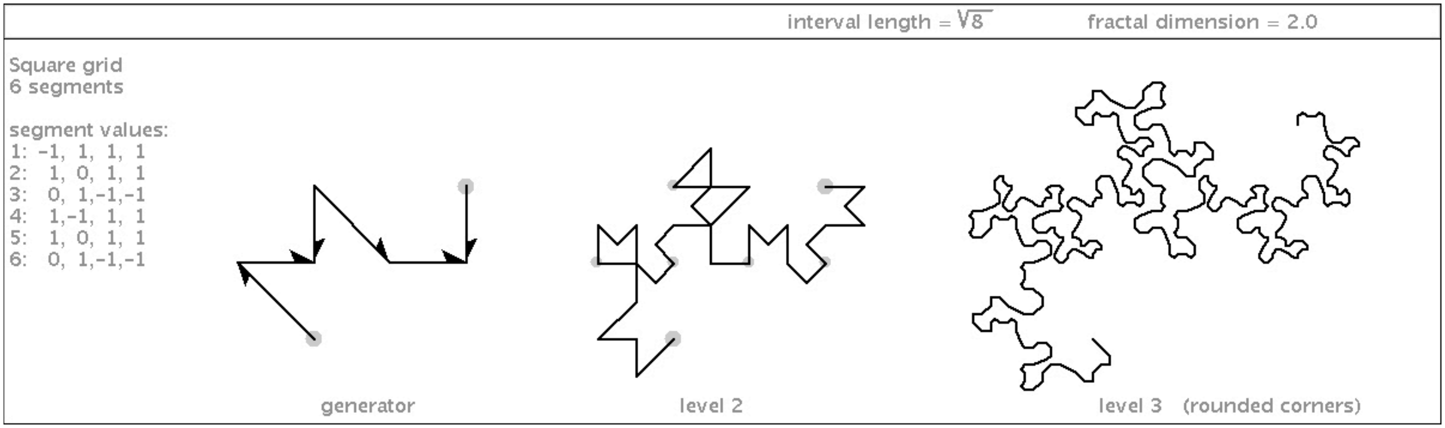


Let's see it rendered at a higher level, below. To the right is the HH Dragon.

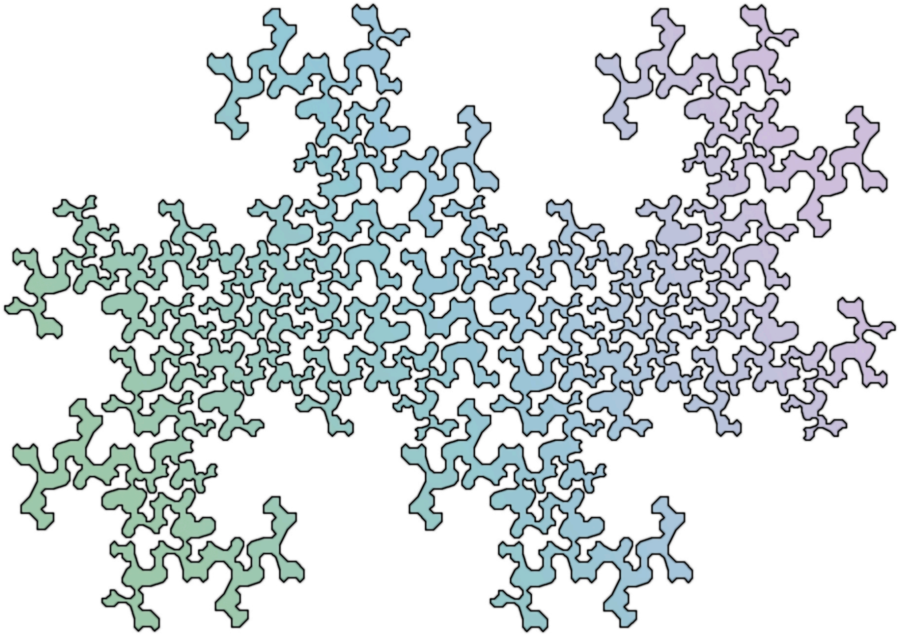




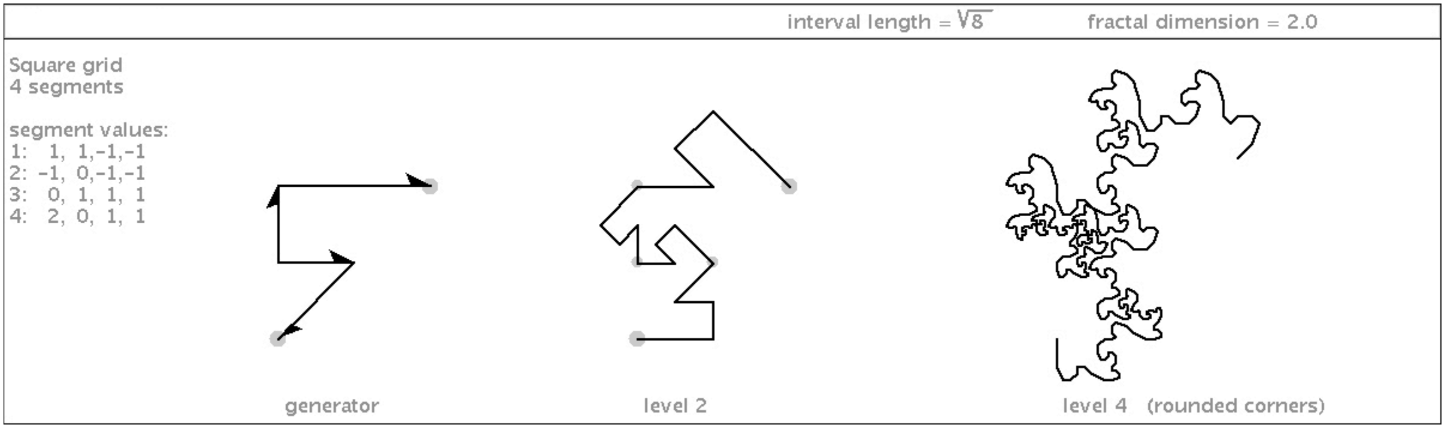
Here’s another specimen that is related to the HH Dragon:

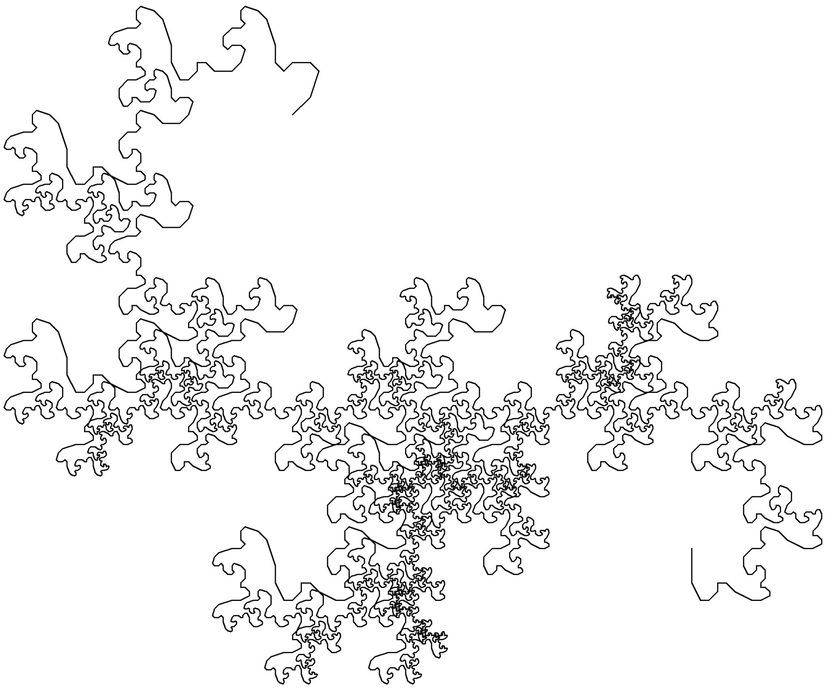


I am impressed with how this curve is so unpredictable and irregular in its internal meandering, yet it is able to avoid any self-crossings (it does self-touch on vertices: those are separated due to the rounded-corners scheme of the drawing). Here is a rendering of two copies of this curve (one flipped 180 degrees). They are combined to make the shape of the twin dragon – which closes the loop, enabling it to be filled internally with color.

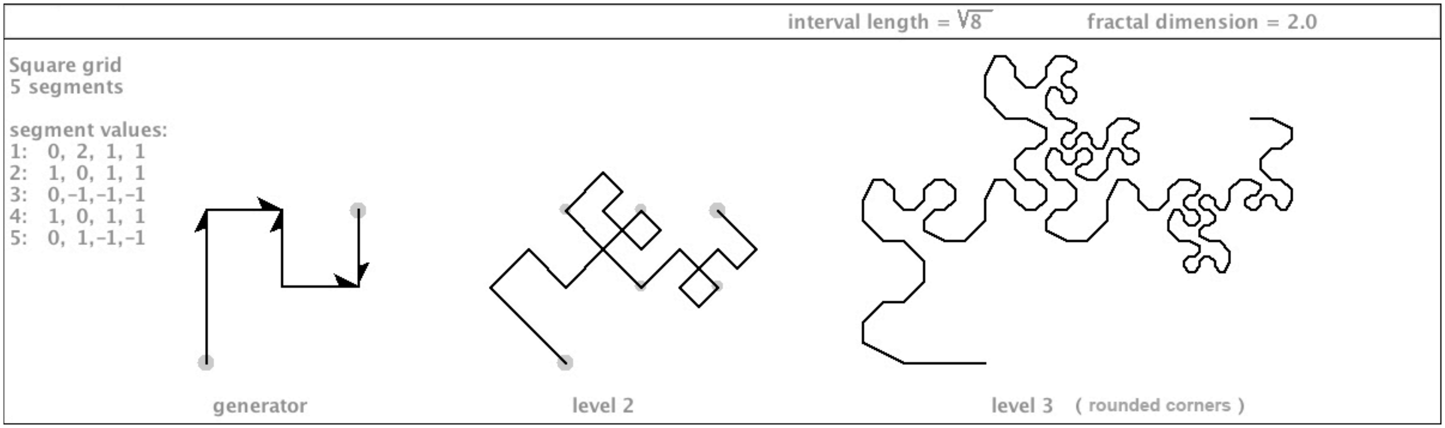


Here's another specimen that resolves to the same shape as the HH Dragon. Since the generator has shorter segments in the middle region, the mid-sections of its tergaons are rather knotted, and full of detail. This specimen's teragons are precariously self-touching (even with rounded-corners). I have rendered it below, rotated 90 degrees.

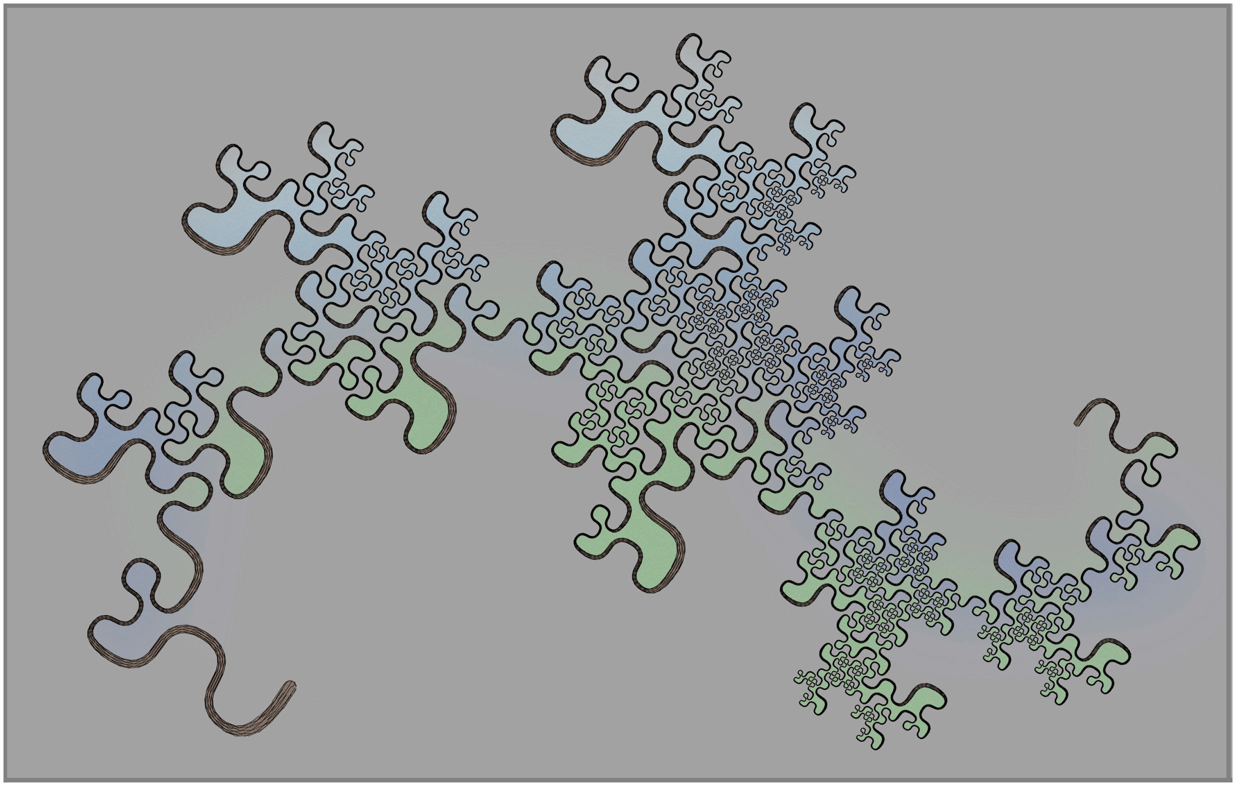




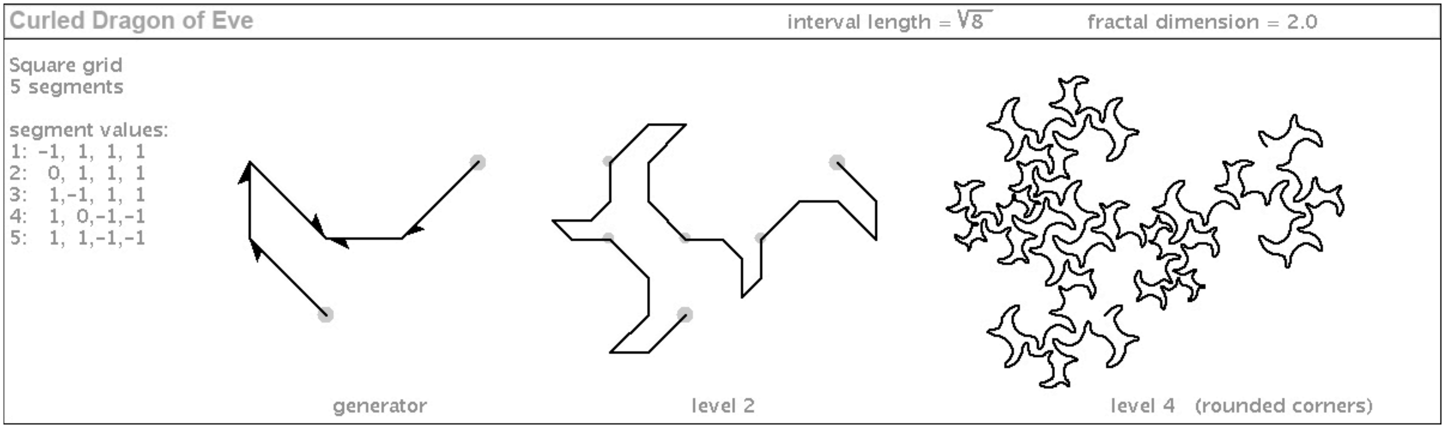
Another √8 Dragon is shown below. This one also resolves to the shape of the HH Dragon.

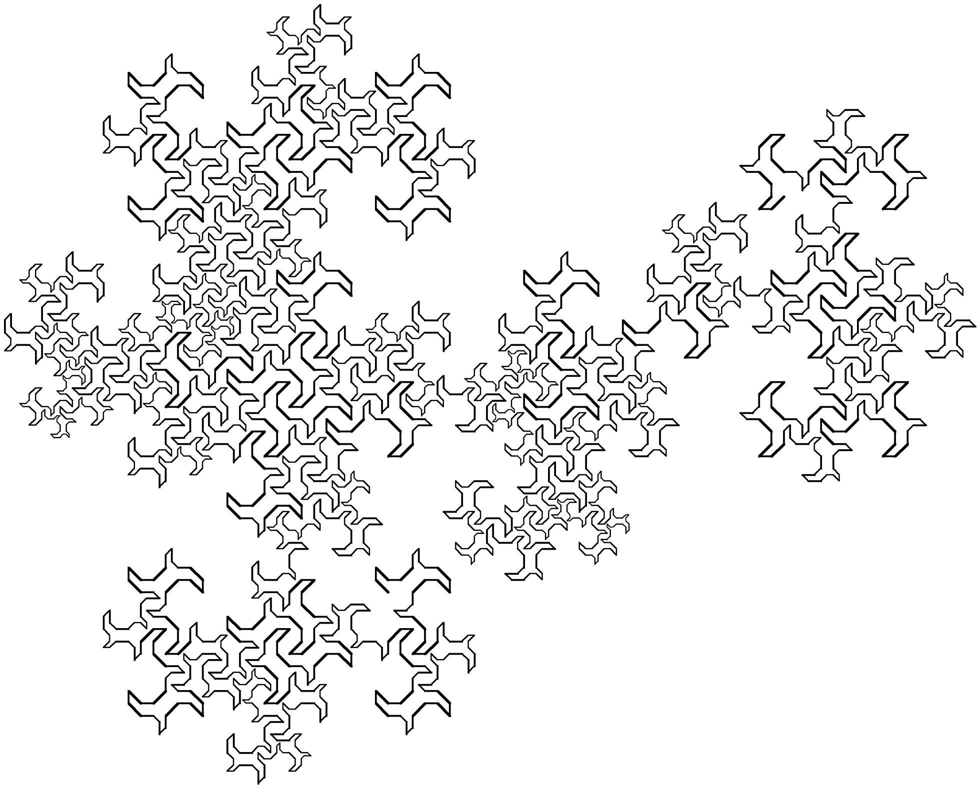


Here it is fractalized at a higher level, splined, colorized, and slightly rotated…for your brain-filling pleasure.

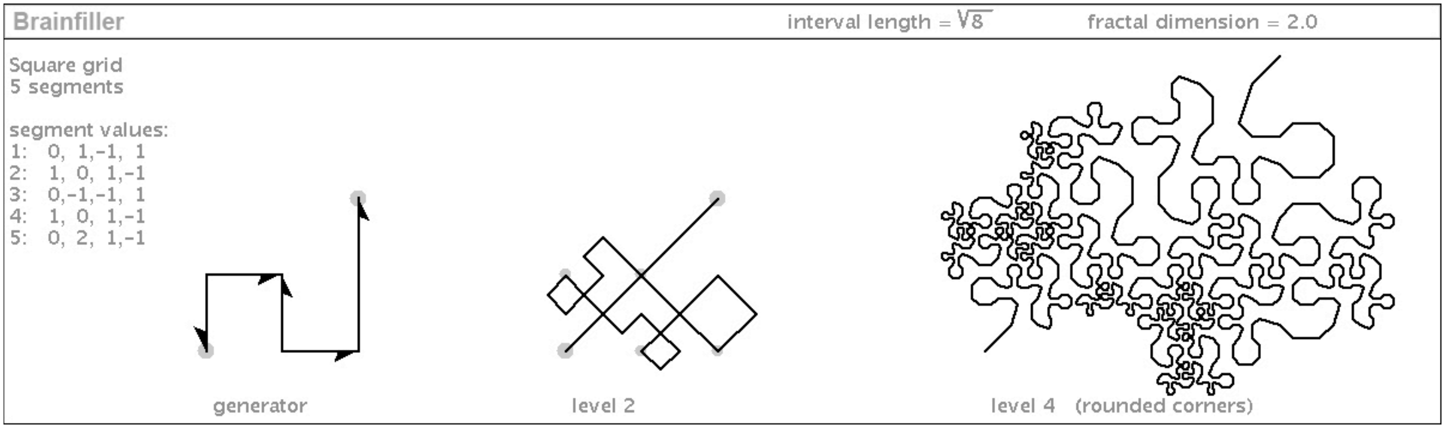


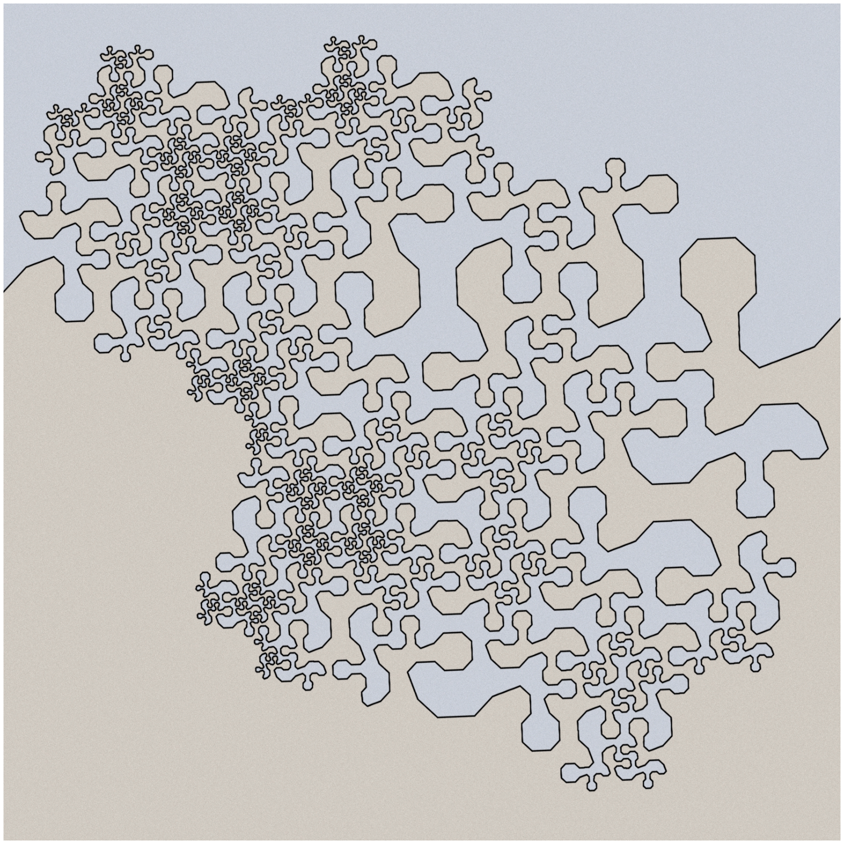
Here is a self-avoiding dragon that I was excited to discover. It appears to be a relative of the *Dragon of Eve*. It's like adding a smaller triangular bump onto the big triangular bump of the Dragon of Eve. It's like a curly Dragon of Eve!





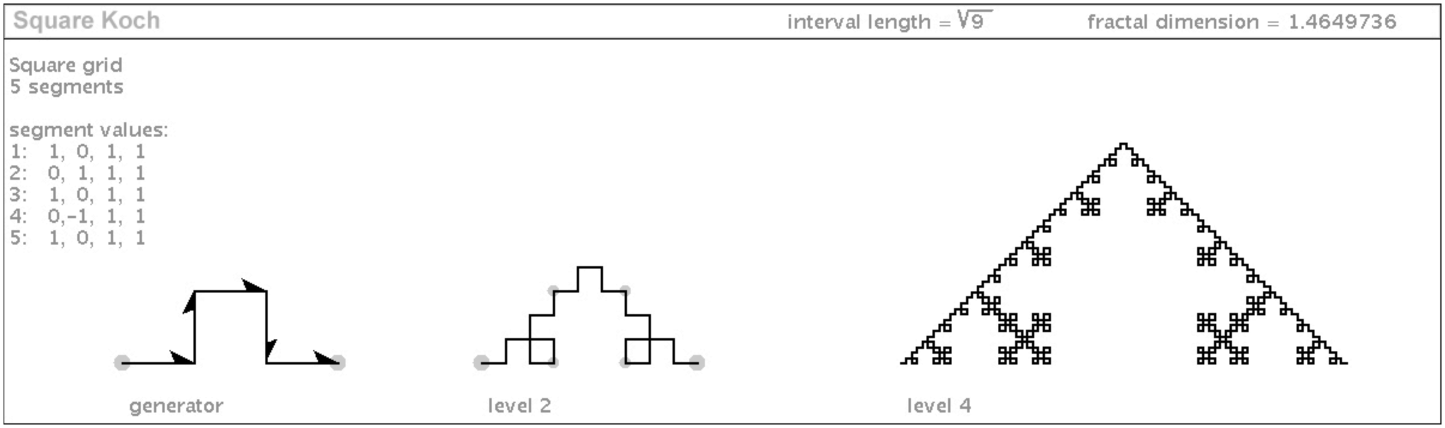
There is one last fractal I want to show in the √8 family: I call it "Brainfiller". Below it is colored and rotated.



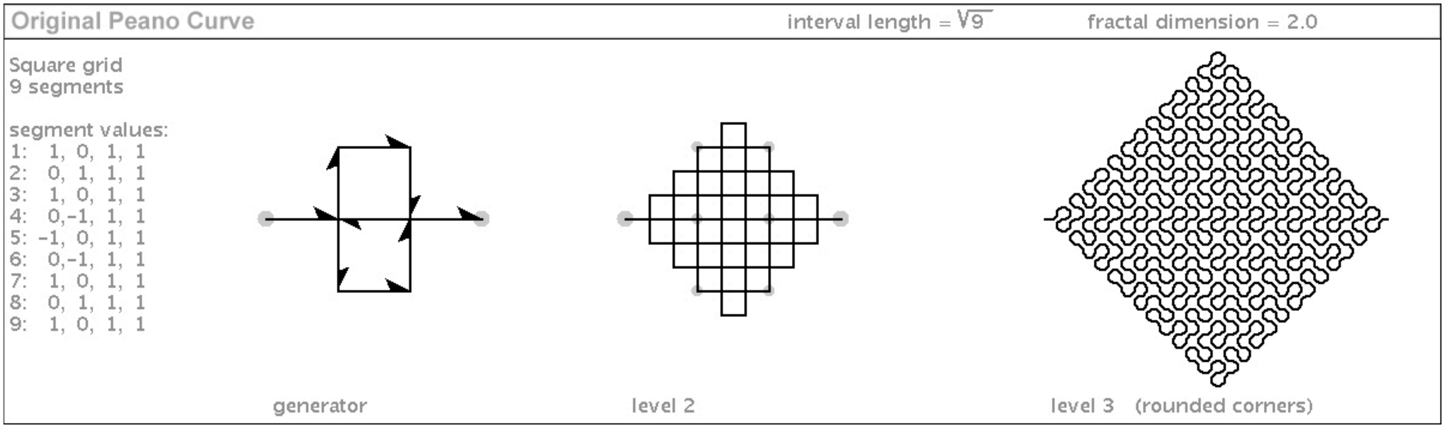


√9

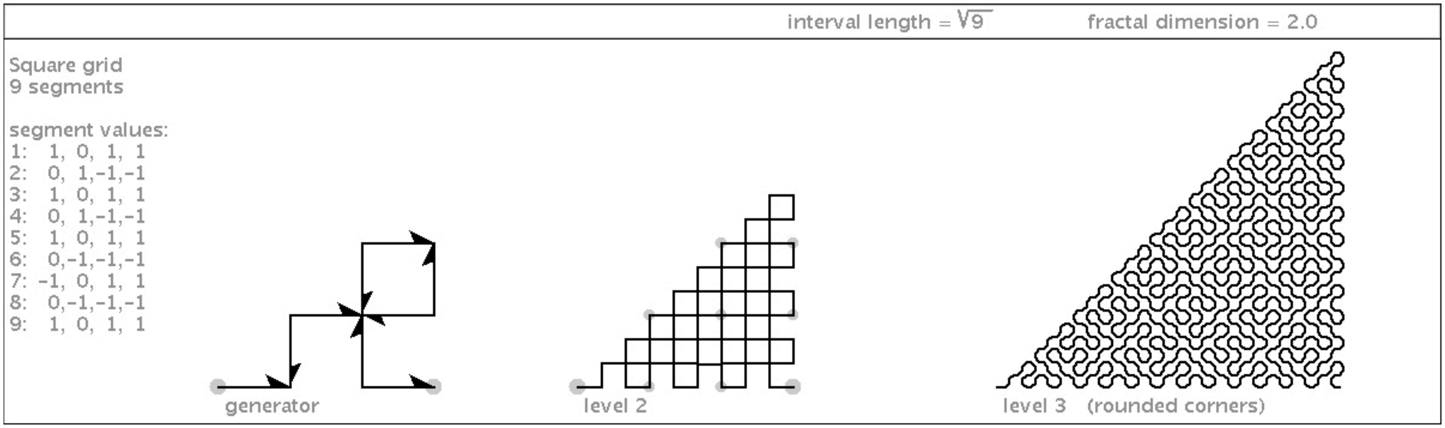
The √9 square grid family has many interesting plane-filling curves. The square root of 9 is of course 3. And, being an integer, my scheme places the interval length horizontal, stretching across three grid cells. We already encountered one member of this family early on when I showed you the Koch curve: its friend, the Square Koch. Here is its diagram:



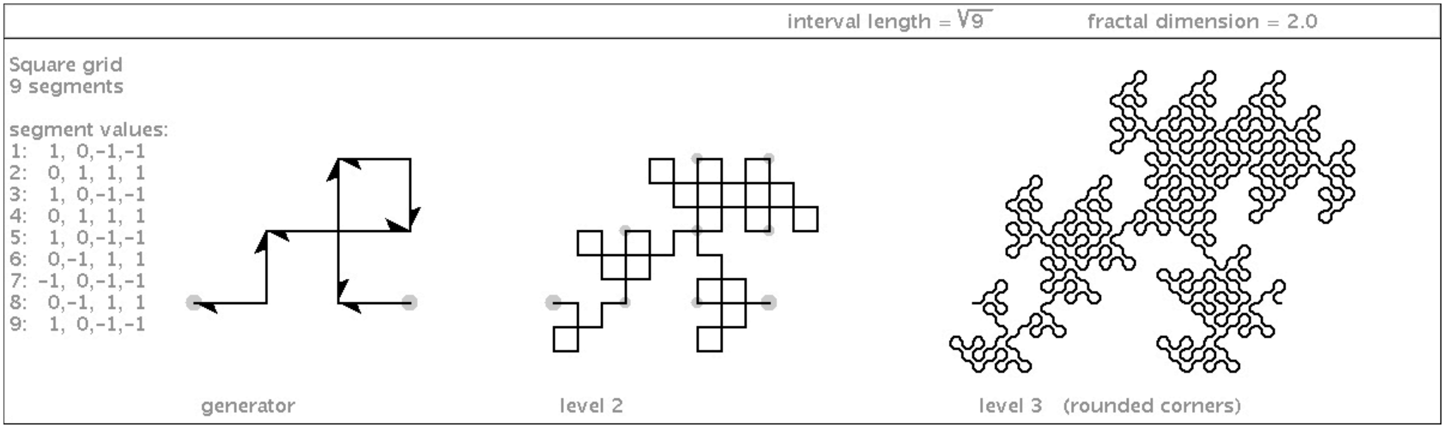
Now let's add a floor and a basement to this generator and see what happens. Lo and behold, we get one of the most familiar – and oldest – plane-filling curves of all time: the "Original Peano Curve" (as Mandelbrot called it: Giuseppe Peano explored many variations). Rounded corners help a lot when viewing this curve.



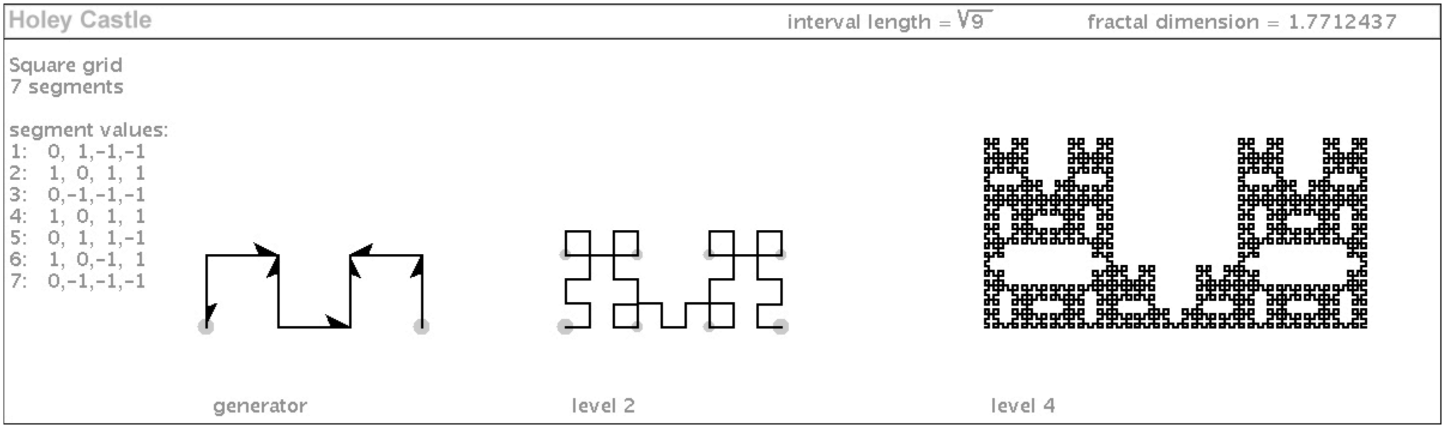
Here is a variation that fills the same area as the original Peano curve, but the shape it fills is a right triangle:



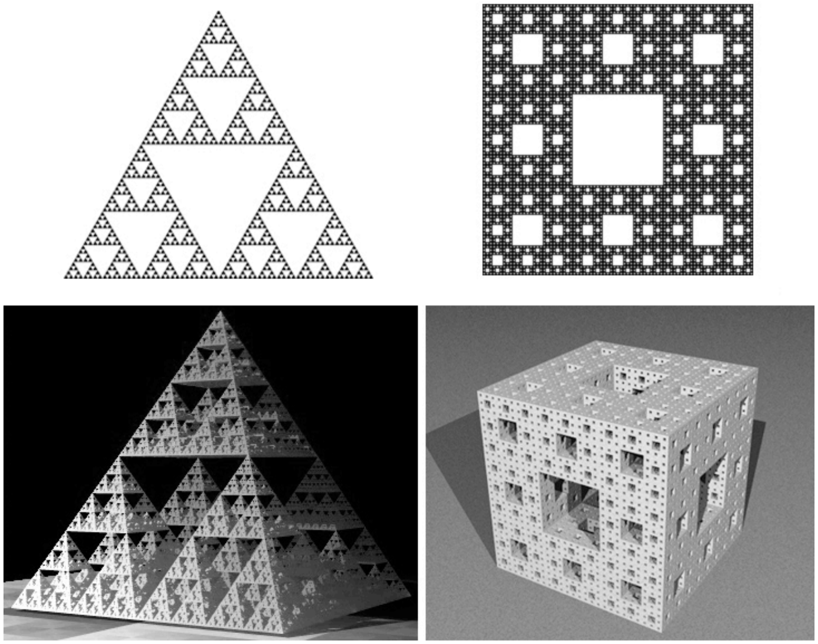
And here is a familiar theme once again: a generator shape can be made to create either a right triangle or a dragon, by way of alternate flippings. This variation is like a dragon (okay, maybe it's not like a dragon… let's just go with "jaggy").



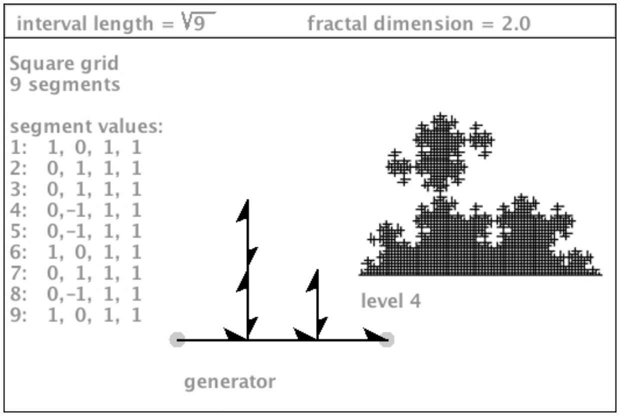
Where there be dragons…there be castles. The √9 Castle below is filled with holes…holes of all sizes. It is a holey castle. No surprise: its fractal dimension is only ~1.77.

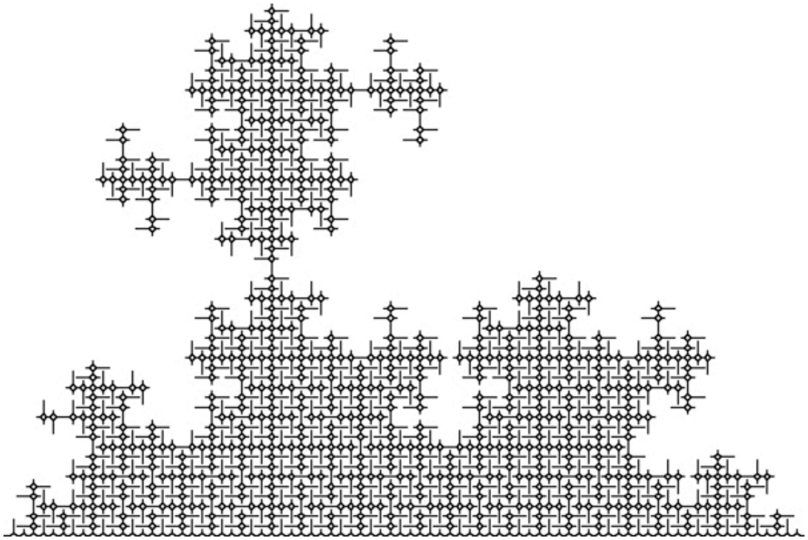


The Holey Castle is related to a large class of fractals that are riddled with holes, such as the Sierpinski Triangle and the *Sierpinski Carpet* (and their 3D counterparts: the Tetrix and the Menger Sponge):

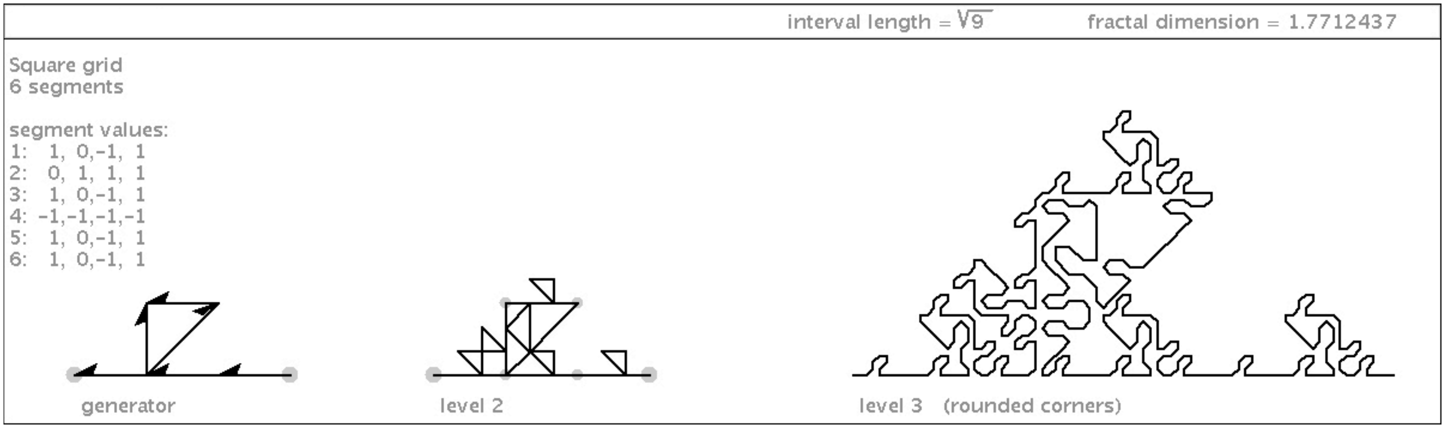


Remember the variation of Cesàro's Sweep I showed you from the √4 family? It has a double-sided vertical needle. Well, I wondered if there might be something similar in the √9 square grid family …and I came up with the generator below. Like Cesàro's Sweep, this curve is everywhere edge-self-touching except for the bottom edge. But unlike Cesàro's Sweep, it has a wonderful fractal boundary. Its 4th teragon is shown below at right. Rounded corners help only slightly to reveal the curve's trajectory.

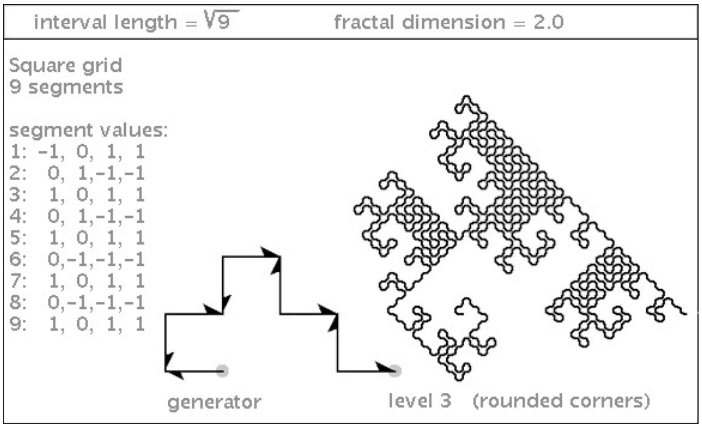


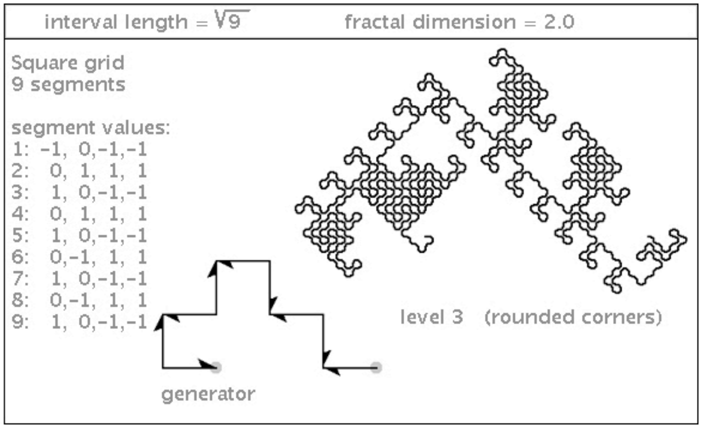


Below is an intriguing fractal curve of dimension ~1.77.

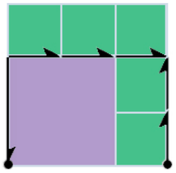


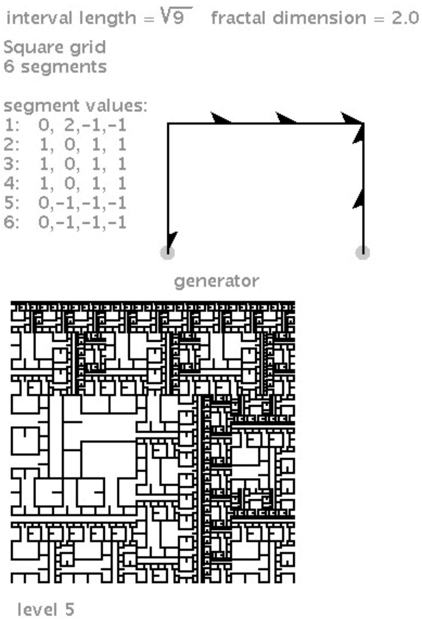
Here are two gridfillers based on a common generator:

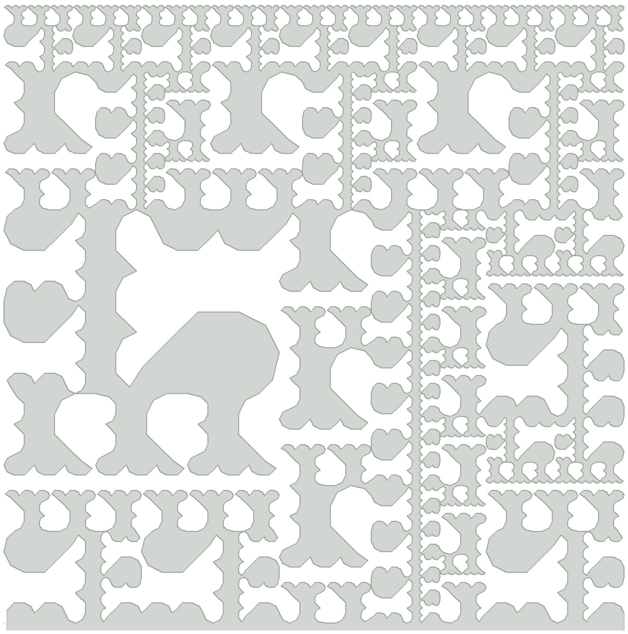


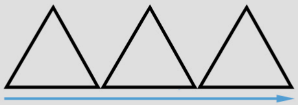


The sepcimen below has a generator with a 2-length segment corresponding to a 2x2 square (shown in purple at right). It has a lot of self-touching edges, and so I used the low-pass smoothing filter to render level 5 with filled-in areas to show the interesting self-similarity.

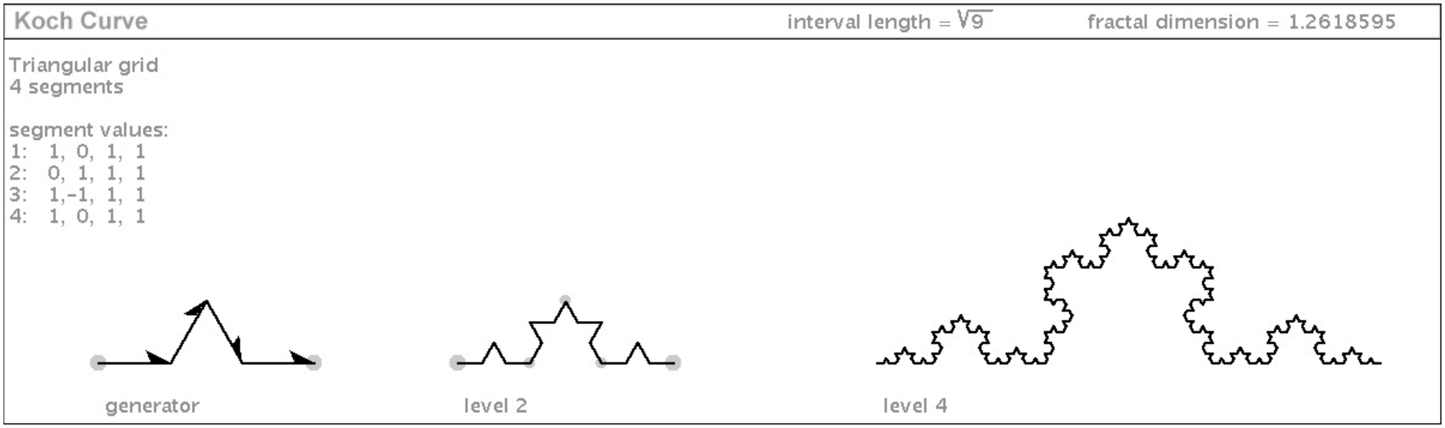




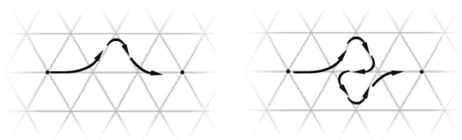


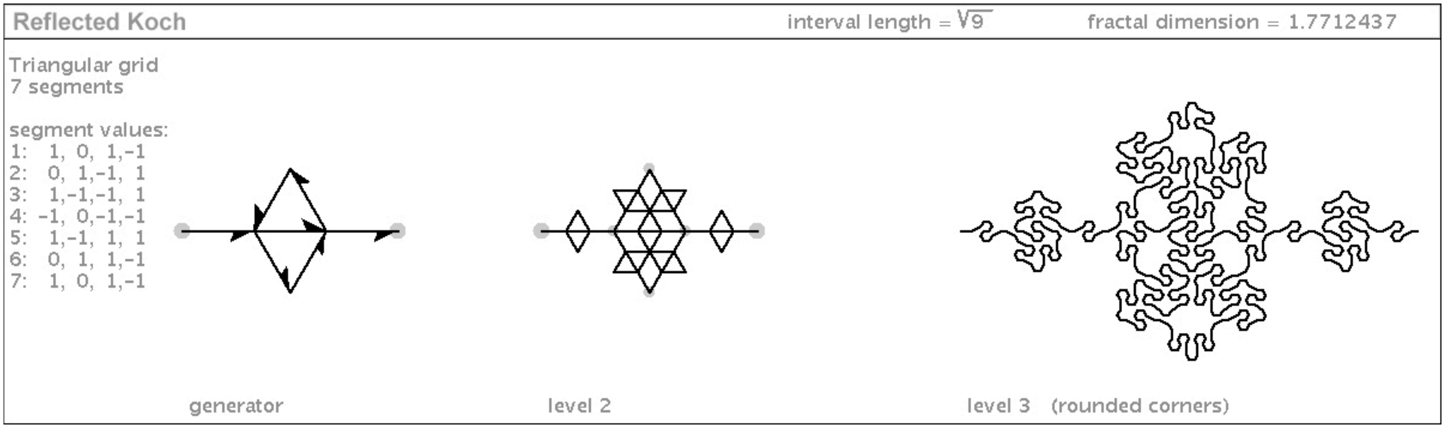
√9 

We are about to explore one of the most versatile, fertile, and abundant families of plane-filling fractal curves: the √9 triangle grid family. The first specimen to mention is none other than…the Koch Curve! Yes, our old friend Koch lives in a triangular grid, and its interval length is 3 units long. The Koch curve has many cousins, and you'll see some similarities.

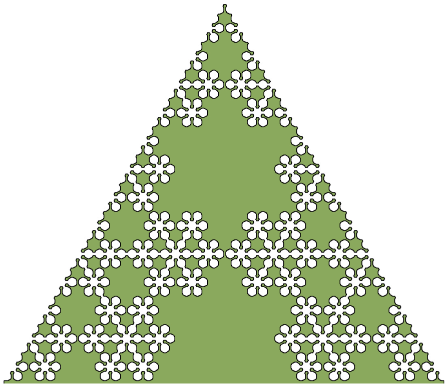


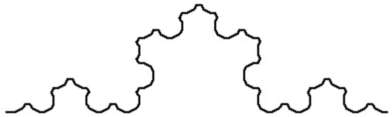
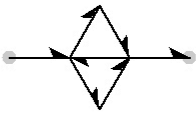
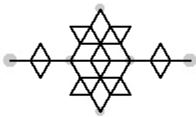
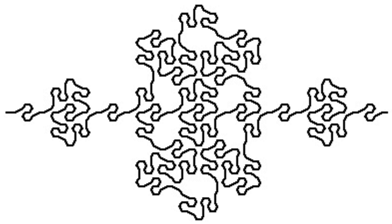
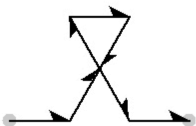
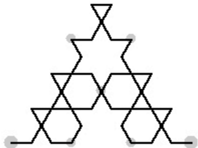
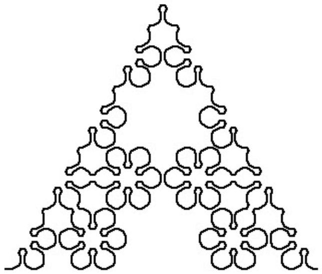
What happens if you draw the first three segments of the Koch Curve and then instead of heading straight to the end, you flip back and draw the remainder as a reflected Koch? Well, here's what you get.



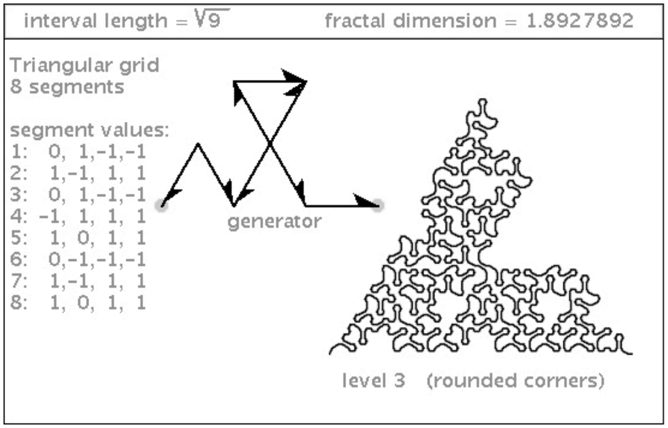


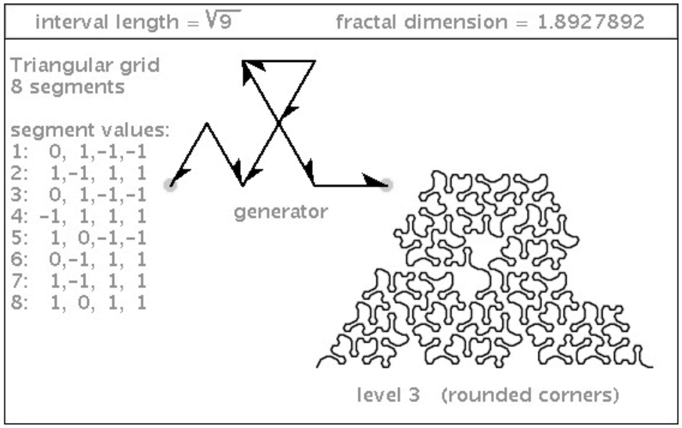
With this curve, which I call “Reflected Koch”, the fractal dimension goes up to ~1.77, since it has 7 segments instead of 4. Below is a table showing these two curves, and one other member of the √9 triangle grid family. You can see that the reflected Koch has obvious similarities to the Koch Curve, having the same profile reflected on top and on the bottom. The curve shown at the bottom does not have the same profile: its generator has a triangle balanced upside-down on top of the Koch bump: it fractalizes upward to make a triangle-shaped conifer tree with a lovely Koch Snowflake inside (shown at right). I call it “Koch Holiday Tree”.



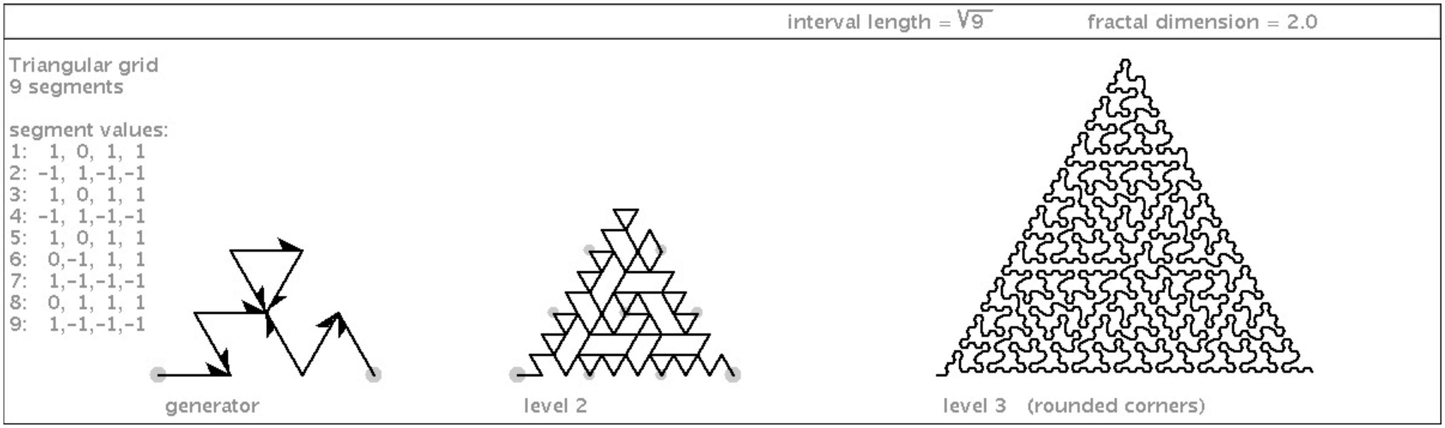
	generator	level 2	level 3 (rounded corners)
interval length = 3 num segments = 4 dimension = 1.2618595			
interval length = 3 num segments = 7 dimension = 1.7712437			
interval length = 3 num segments = 7 dimension = 1.7712437			

Now, taking the Koch Holiday Tree generator, let's add a triangular bump at the left, and try a few variations in flippings. Here are two results:

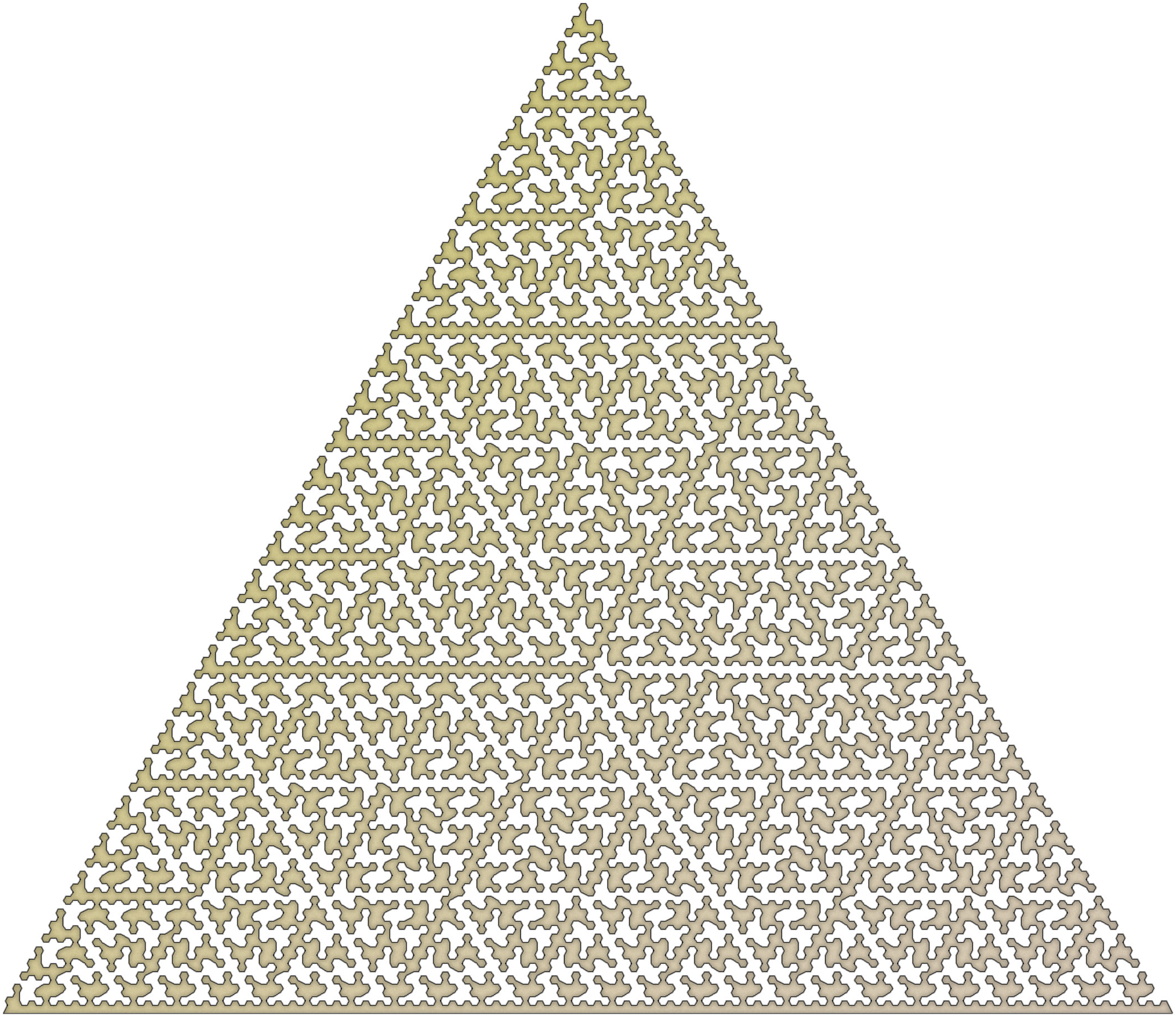




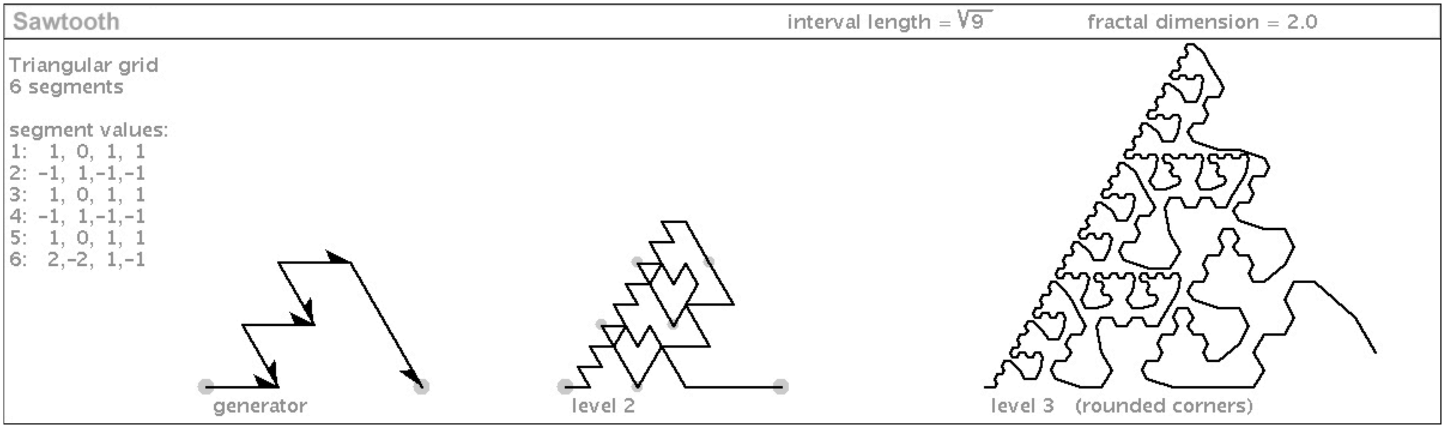
Notice that the fractal dimension has gone up to ~1.89. It seems we are getting close to filling a triangle, but there are still holes. One more variation, with some clever flippings brings the total number of segments to 9. Tada – we have a plane-filling curve!

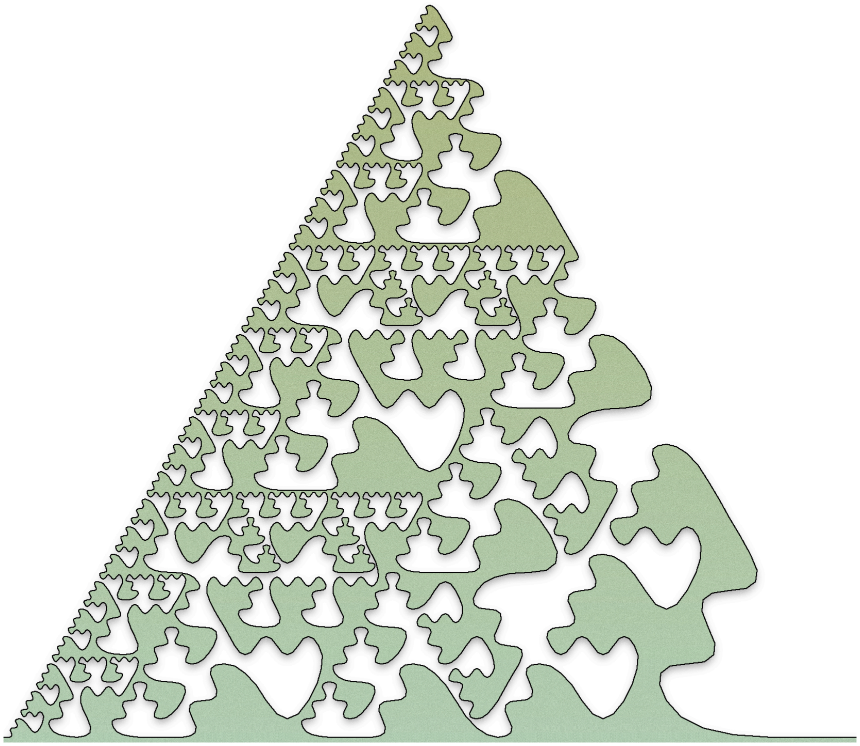


This is shown in higher resolution with rounded corners and filled-in, on the next page.

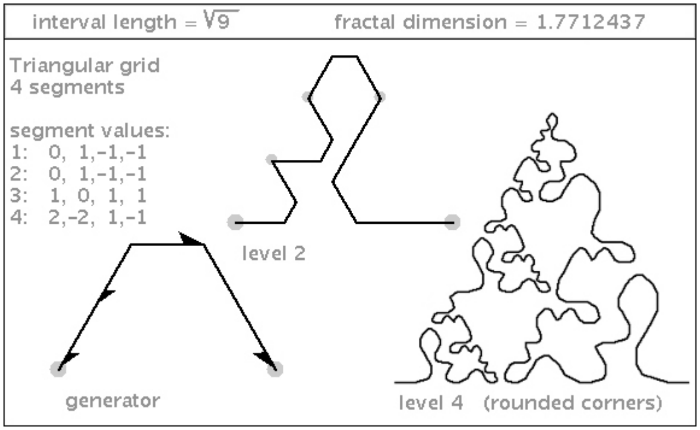


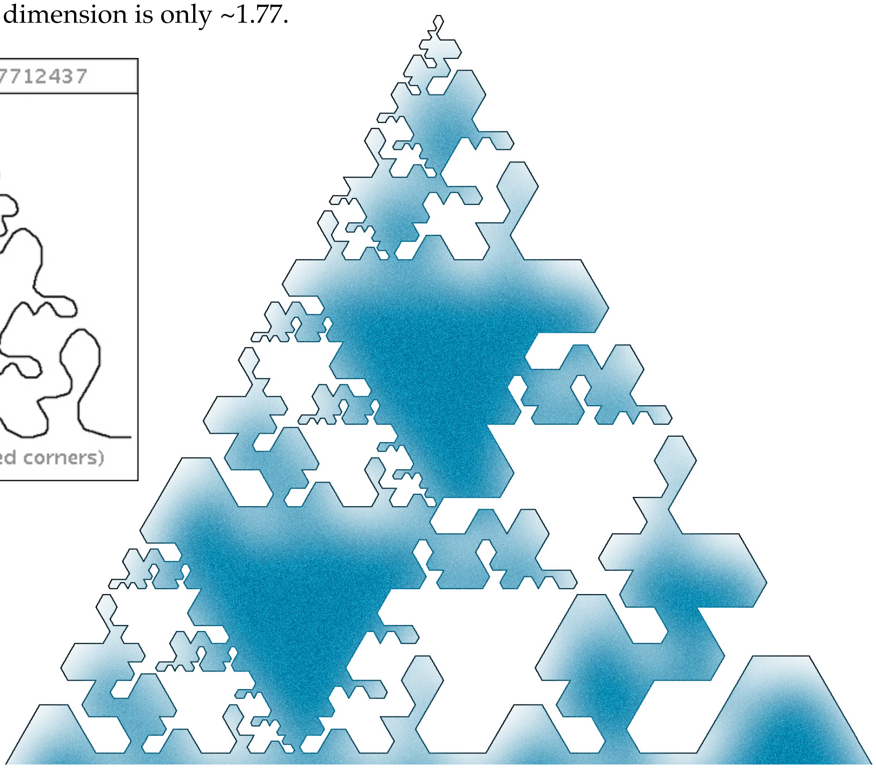
Can a generator with fewer than 9-segment be used to fill a triangle? Sure thing. But at least one segment has to be longer than 1. This next specimen has a generator with 6 segments, and one of those segments has a length of √3. I call it “Sawtooth”. As you would expect, the process by which the triangle gets filled is less regular than with the last specimen.



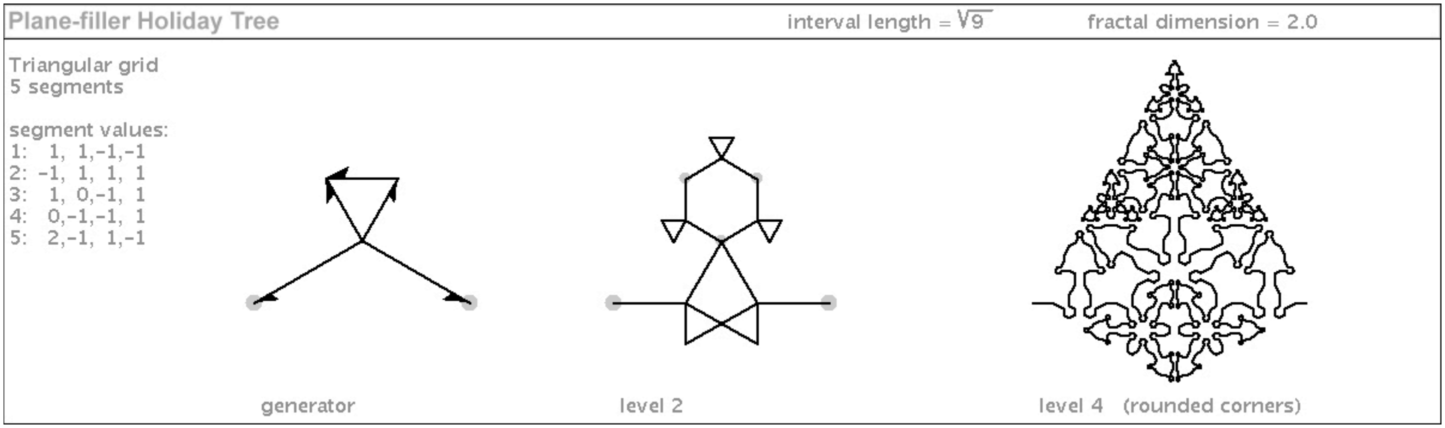


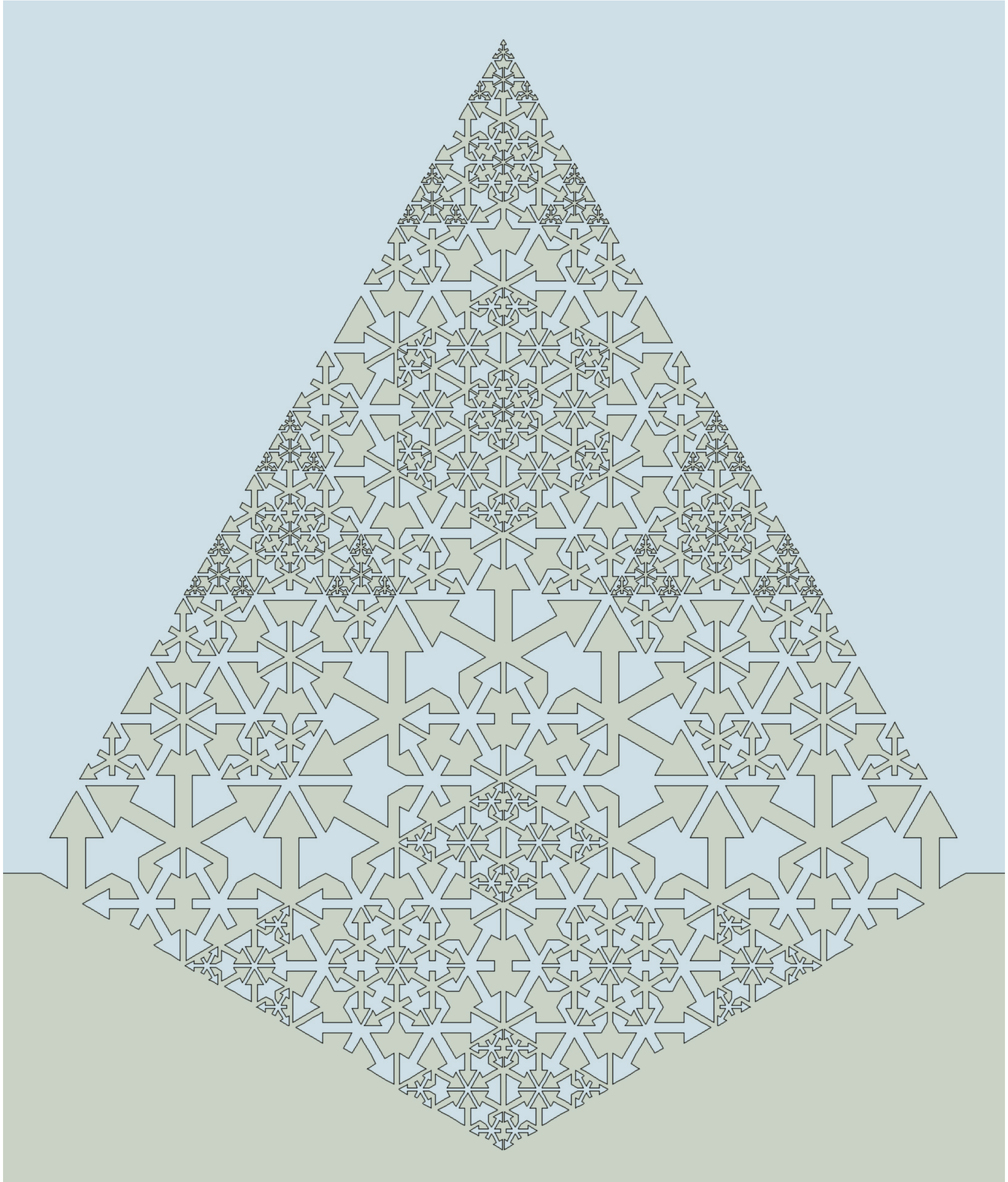
This specimen cannot plane-fill the triangle: its dimension is only ~1.77.



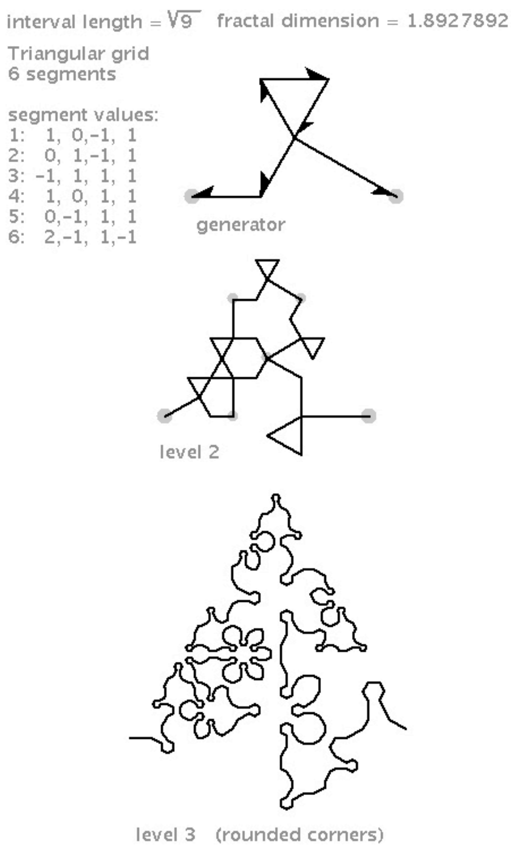


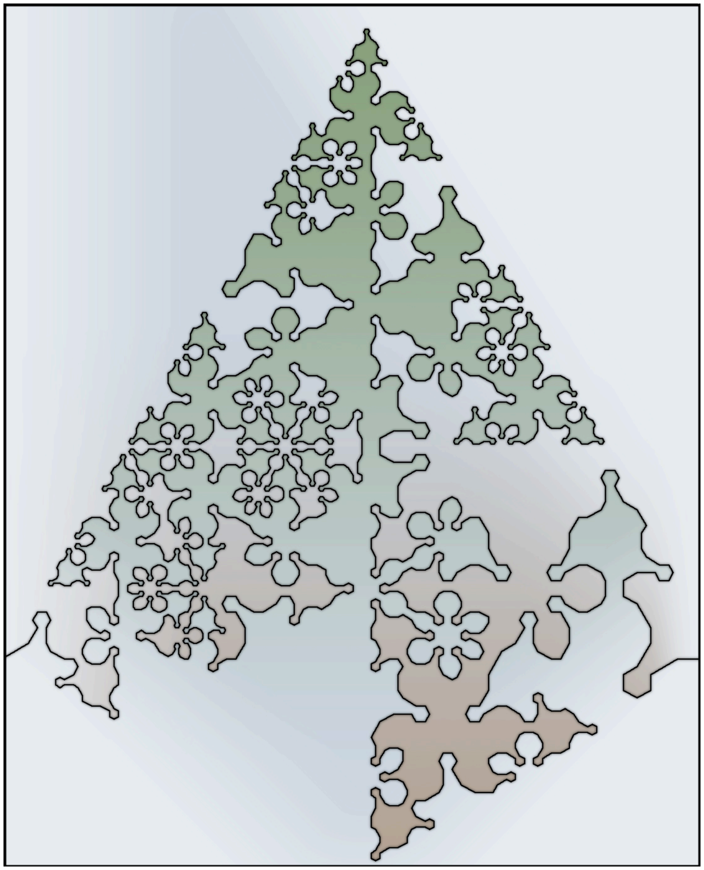
This specimen is a plane-filler. It is a relative of the Koch Holiday Tree I showed you earlier. I call it "Plane-filler Holiday Tree". Strangely enough, this one fills the triangle but it also dips down into the soil – perhaps to nourish its leaves.



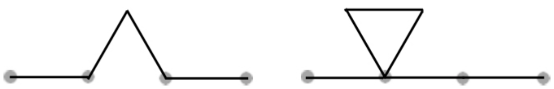


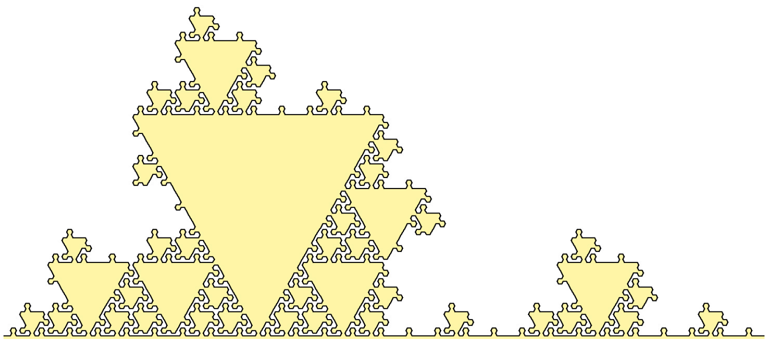
This one is related to the plane-filler Holiday tree. It has a stylish asymmetric flair.

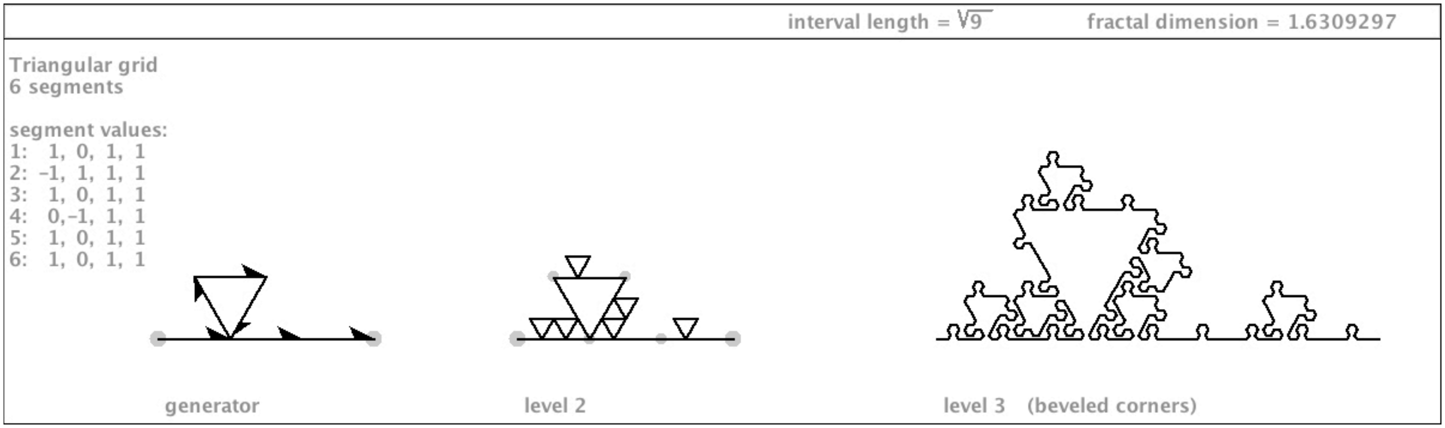




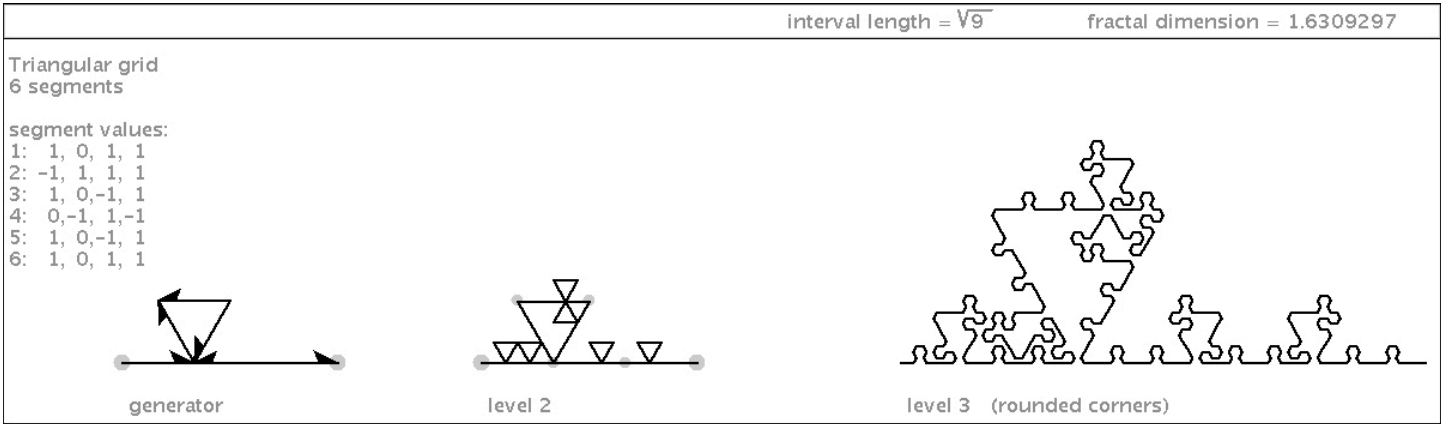
Now let's come back to the Koch Curve and explore a variation of the generator in which the triangular bump is turned unto a full triangle, and tilted up on its vertex, to make the following generator:



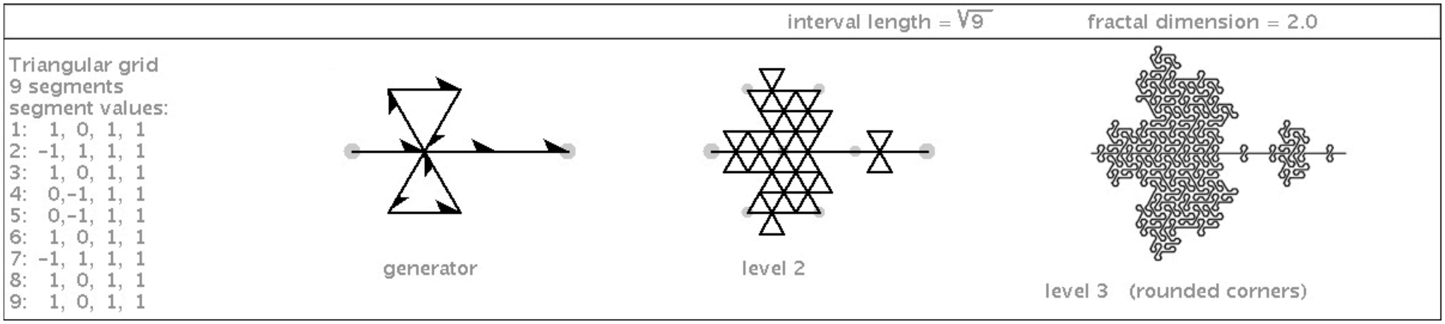




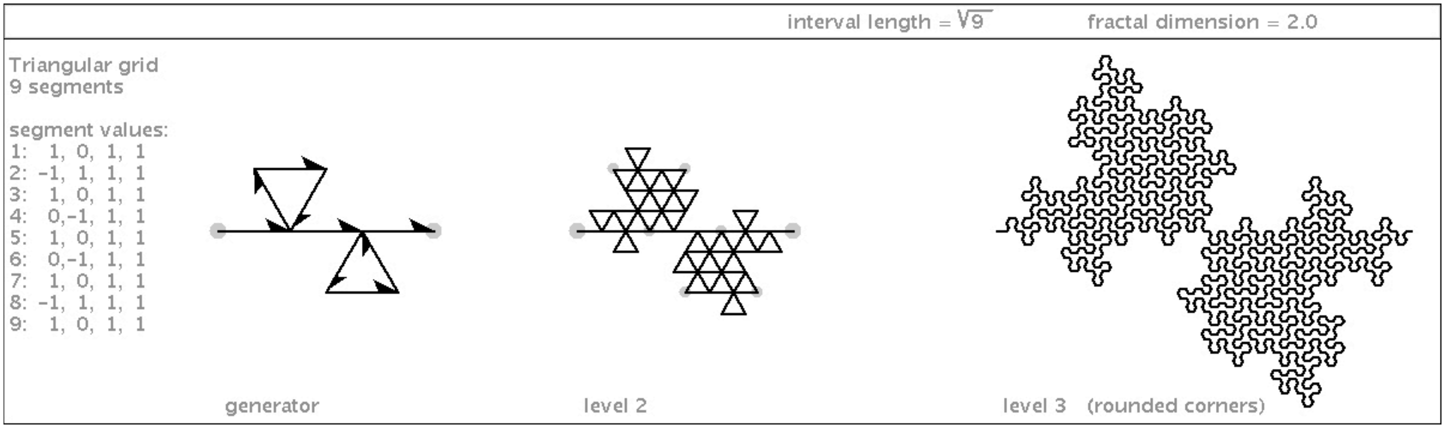
Here's what happens if we use the same generator shape and fiddle with the flippings:



Now, if we make a reflection of this triangle underneath the horizontal line, and use no flippings, we have a gridfiller!

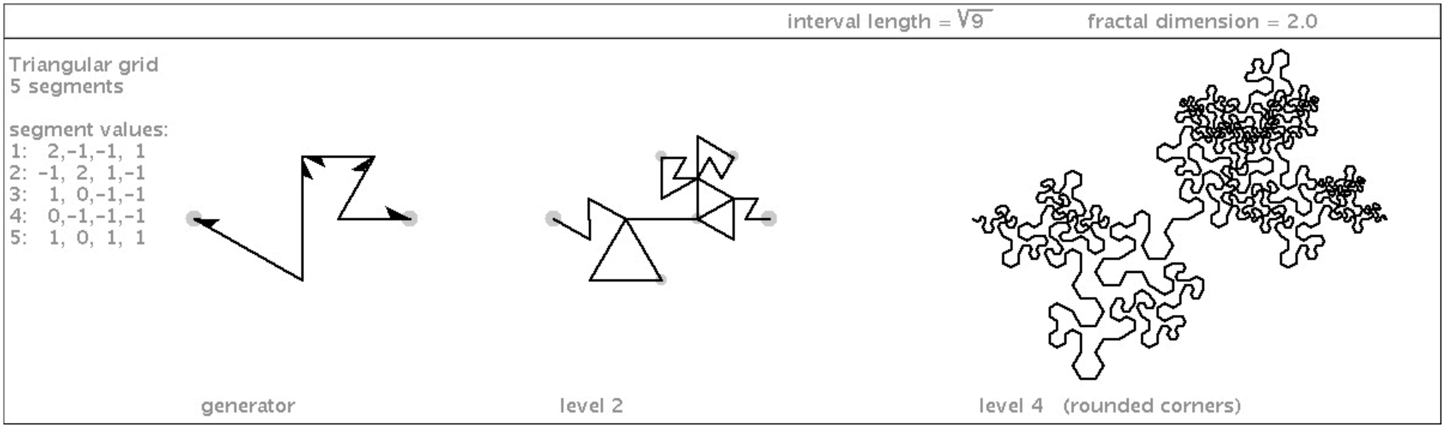


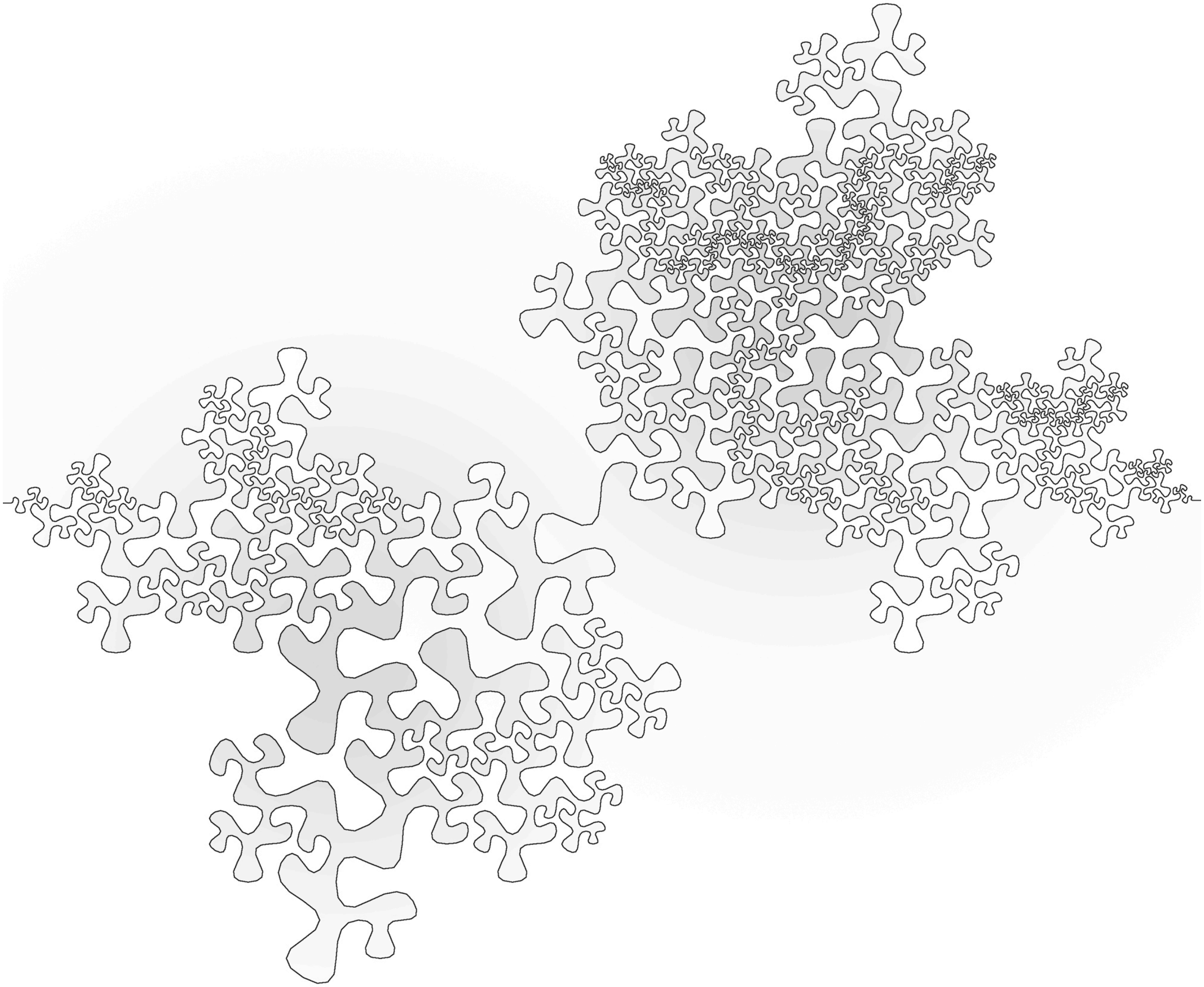
And, what if we shift that lower triangle to the right? Here’s what we get:



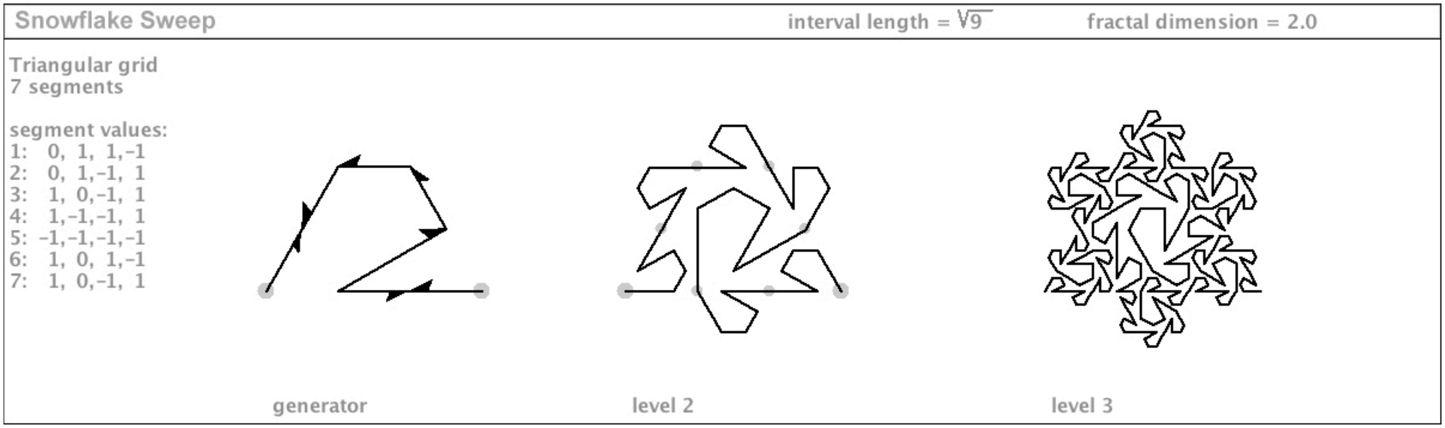
Did you notice something? This has the same shape as the pinched Ter-Dragon of the √3 family we met earlier. You can see that this generator is made up of three copies of the Ter-Dragon, but they are not arranged in the usual Ter-Dragon way.

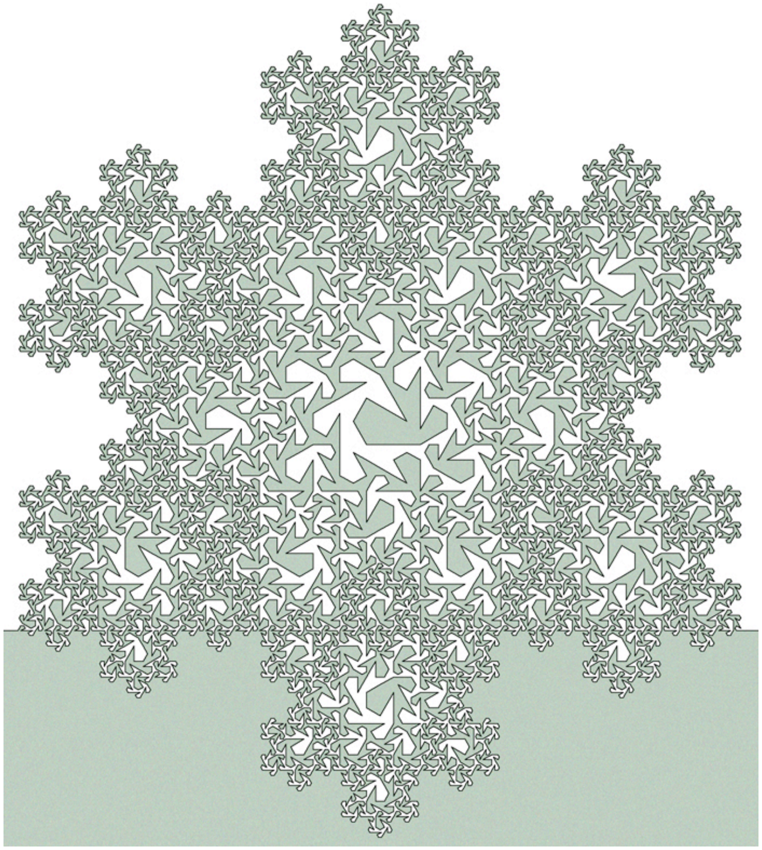
Now check out this variation of the specimen, shown below. It is rendered with splines on the next page.





The √9 triangle grid family includes one of the finest specimens of all: the *Snowflake Sweep*, featured in Mandelbrot's book:

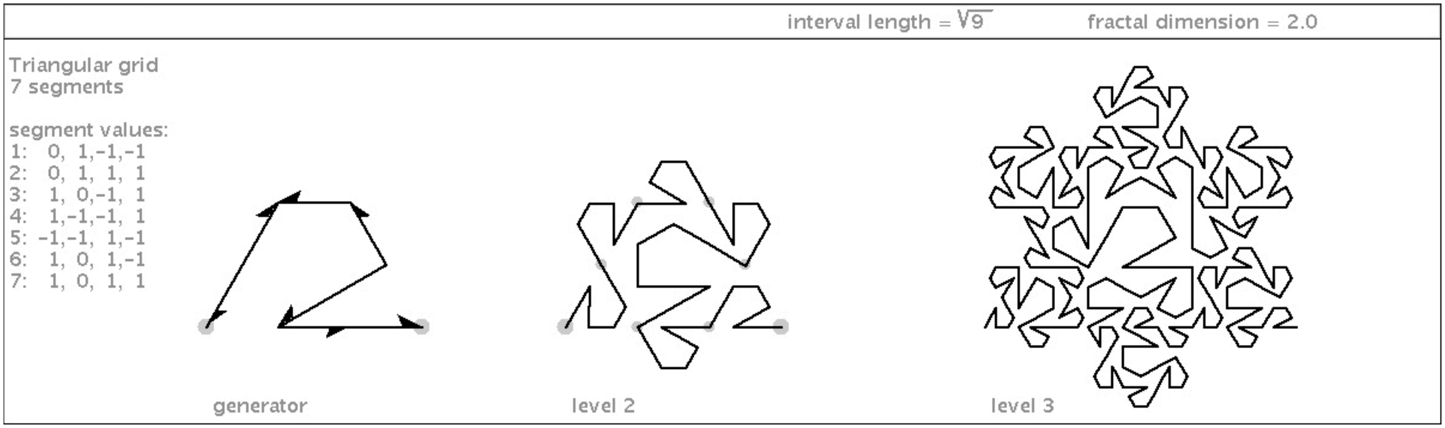




Notice how it fills the Koch snowflake.

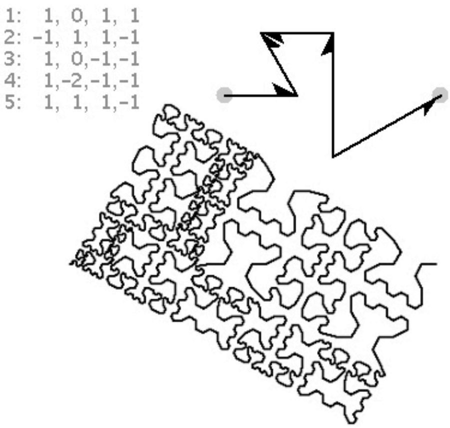
Earlier I pointed out that Mandelbrot had said he "designed" a self-avoiding curve; I suggested that the curve could just as easily have been "discovered". Well, Mandelbrot's Snowflake Sweep is so amazing and beautiful that I can hardly blame him for claiming to be its designer.

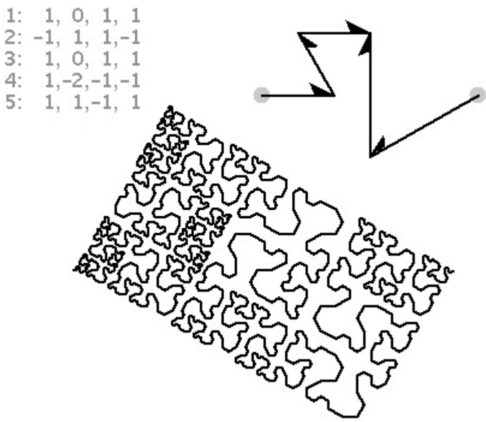
By the way, this curve is very flexible. Any segment in the generator can have an arbitrary flipping in its x value, and the teragons will still be self-avoiding. On the next page is a variation in x-flippings.

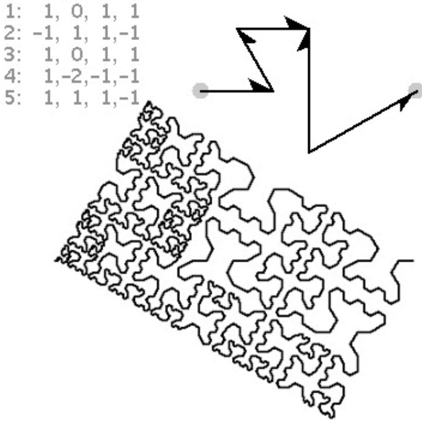


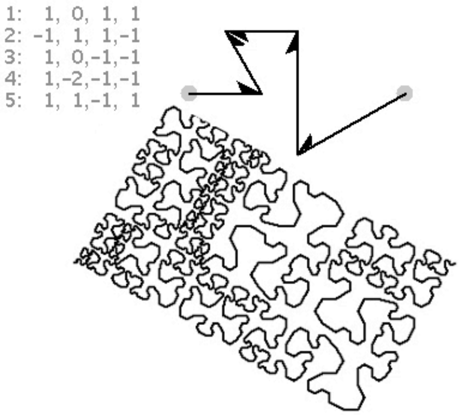
√9 Carpets

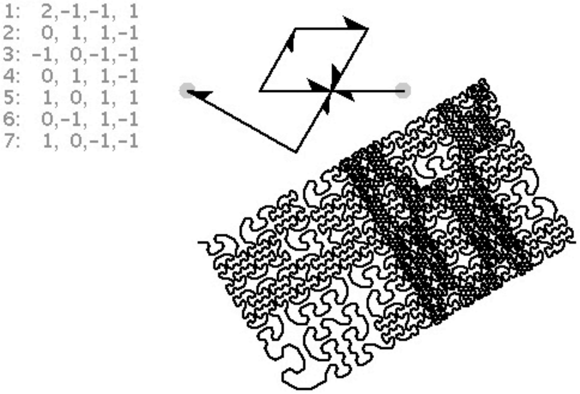
Now I want to show you a series of √9 triangle grid family curves that incorporate segments of length √3, and fractalize to fill a tilted rectangle. I call them "√9 Carpets". I have uncovered several of them. Here is a small selection, for your brain-filling pleasure. In all cases, the 4th or 5th teragons are shown with rounded corners.

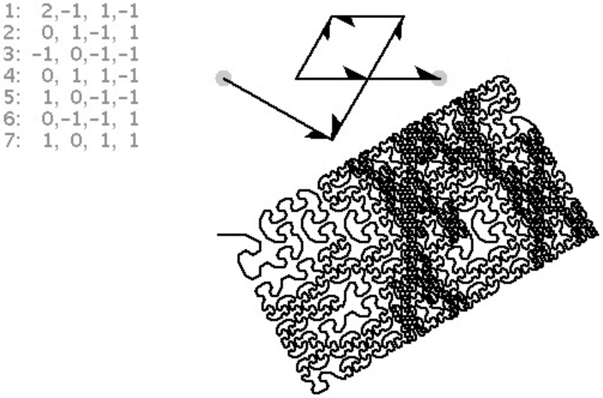


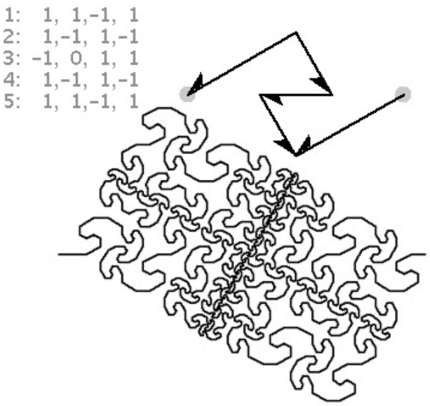


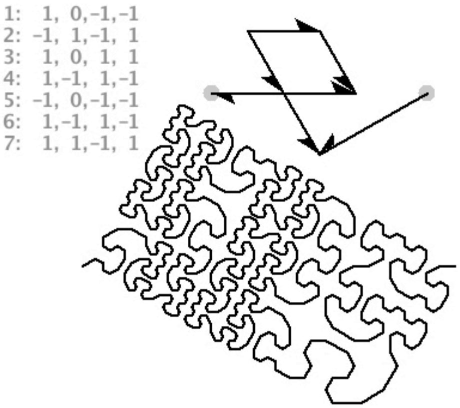

1: 1, 0, 1, 1
2: -1, 1, 1,-1
3: 1, 0, 1, 1
4: 1,-2,-1,-1
5: 1, 1, 1,-1


1: 1, 0, 1, 1
2: -1, 1, 1,-1
3: 1, 0,-1,-1
4: 1,-2,-1,-1
5: 1, 1,-1, 1

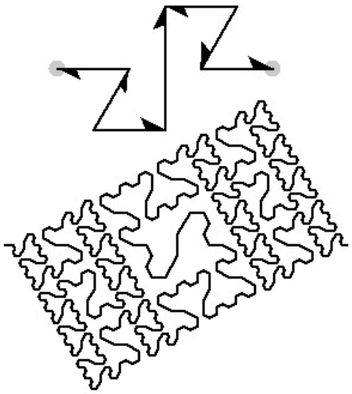

1: 2,-1,-1, 1
2: 0, 1, 1,-1
3: -1, 0,-1,-1
4: 0, 1, 1,-1
5: 1, 0, 1, 1
6: 0,-1, 1,-1
7: 1, 0,-1,-1


1: 2,-1, 1,-1
2: 0, 1,-1, 1
3: -1, 0,-1,-1
4: 0, 1, 1,-1
5: 1, 0,-1,-1
6: 0,-1,-1, 1
7: 1, 0, 1, 1

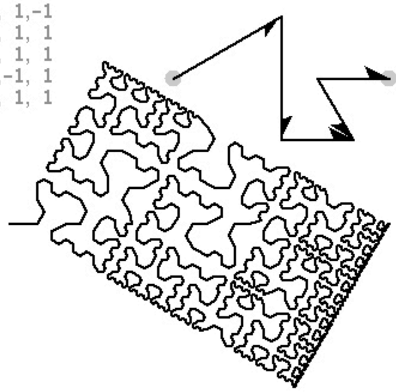

1: 1, 1,-1, 1
2: 1,-1, 1,-1
3: -1, 0, 1, 1
4: 1,-1, 1,-1
5: 1, 1,-1, 1


1: 1, 0,-1,-1
2: -1, 1,-1, 1
3: 1, 0, 1, 1
4: 1,-1, 1,-1
5: -1, 0,-1,-1
6: 1,-1, 1,-1
7: 1, 1,-1, 1

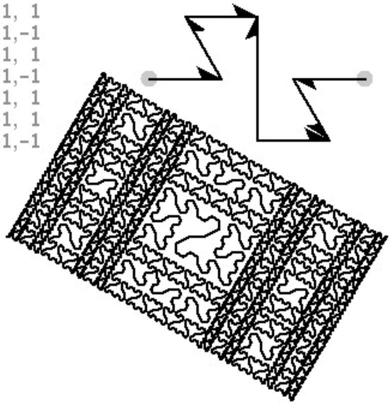
1: 1, 0,-1,-1
2: 0,-1,-1, 1
3: 1, 0, 1, 1
4: -1, 2, 1, 1
5: 1, 0,-1,-1
6: 0,-1, 1,-1
7: 1, 0, 1, 1



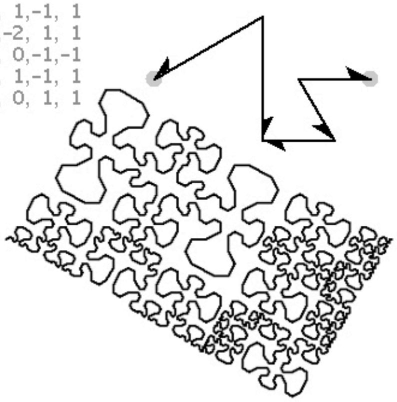
1: 1, 1, 1,-1
2: 1,-2, 1, 1
3: 1, 0, 1, 1
4: -1, 1,-1, 1
5: 1, 0, 1, 1



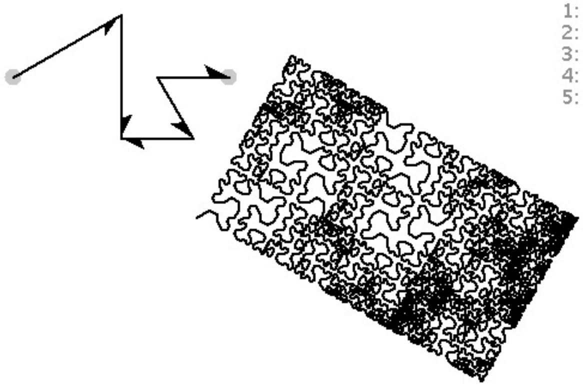
1: 1, 0, 1, 1
2: -1, 1, 1,-1
3: 1, 0, 1, 1
4: 1,-2,-1,-1
5: 1, 0, 1, 1
6: -1, 1,-1, 1
7: 1, 0,-1,-1



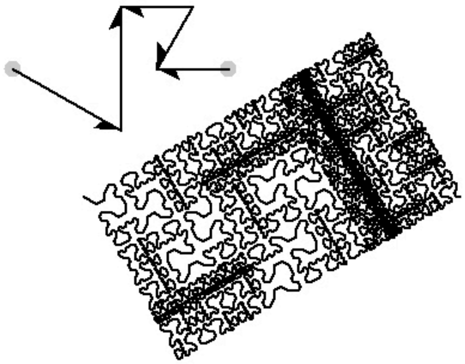
1: 1, 1,-1, 1
2: 1,-2, 1, 1
3: 1, 0,-1,-1
4: -1, 1,-1, 1
5: 1, 0, 1, 1



1: 1, 1, 1,-1
2: 1,-2, 1, 1
3: 1, 0,-1,-1
4: -1, 1,-1, 1
5: 1, 0, 1, 1

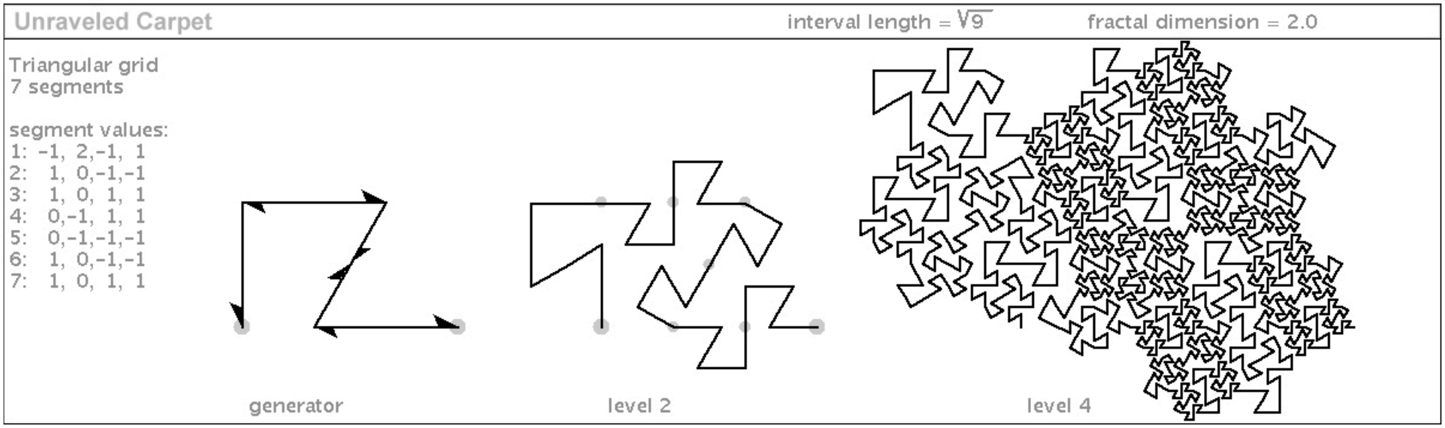


1: 2,-1, 1,-1
2: -1, 2, 1, 1
3: 1, 0,-1,-1
4: 0,-1, 1,-1
5: 1, 0,-1,-1

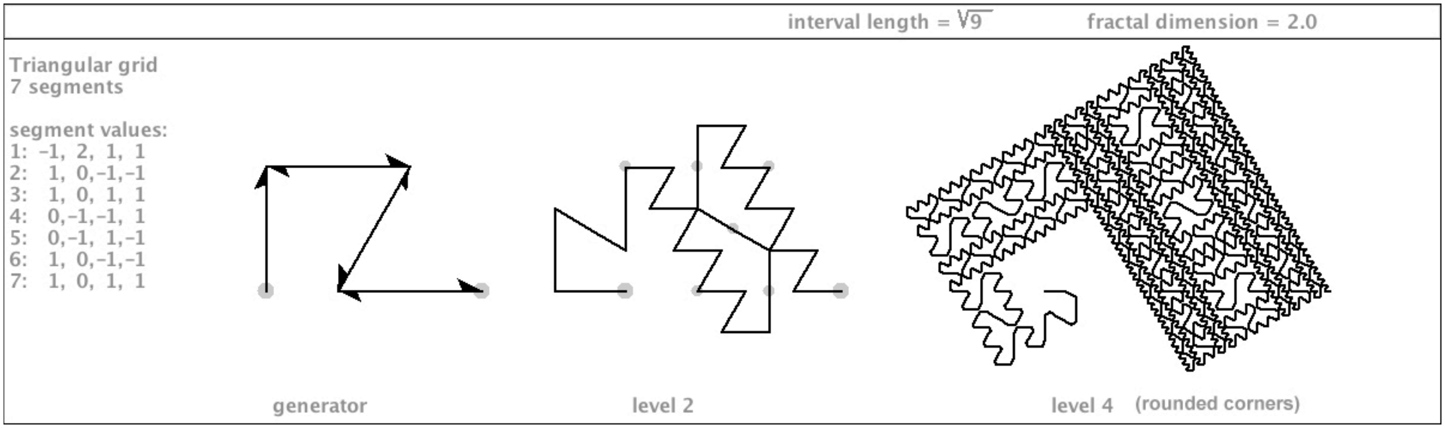


Unraveled Carpets

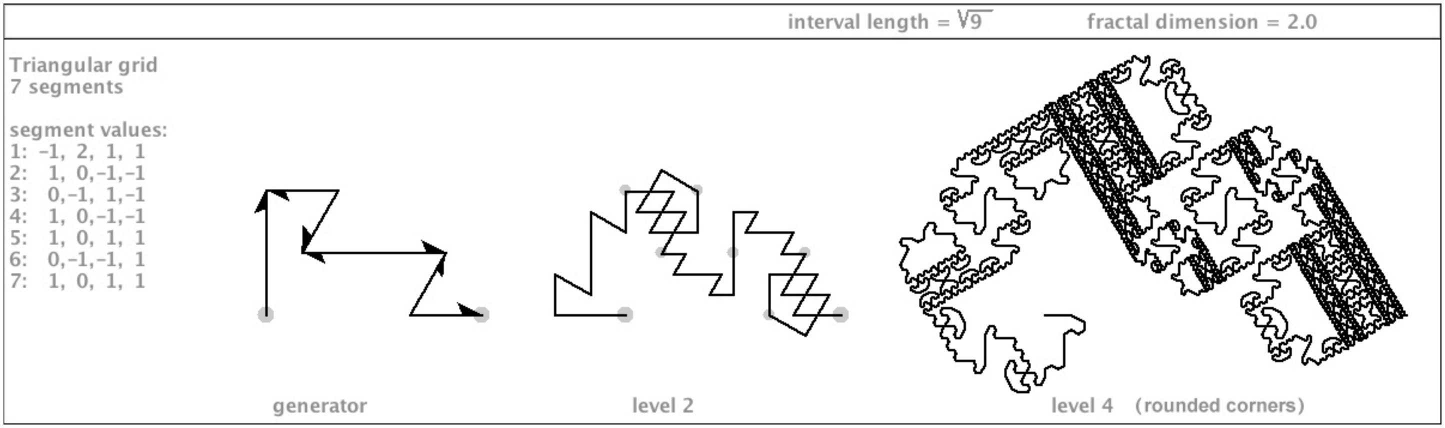
I showed you the following curve early on in the book. This curve seems to be related to the carpets series, except that the weaving has come undone. I call it “Unraveled Carpet”.

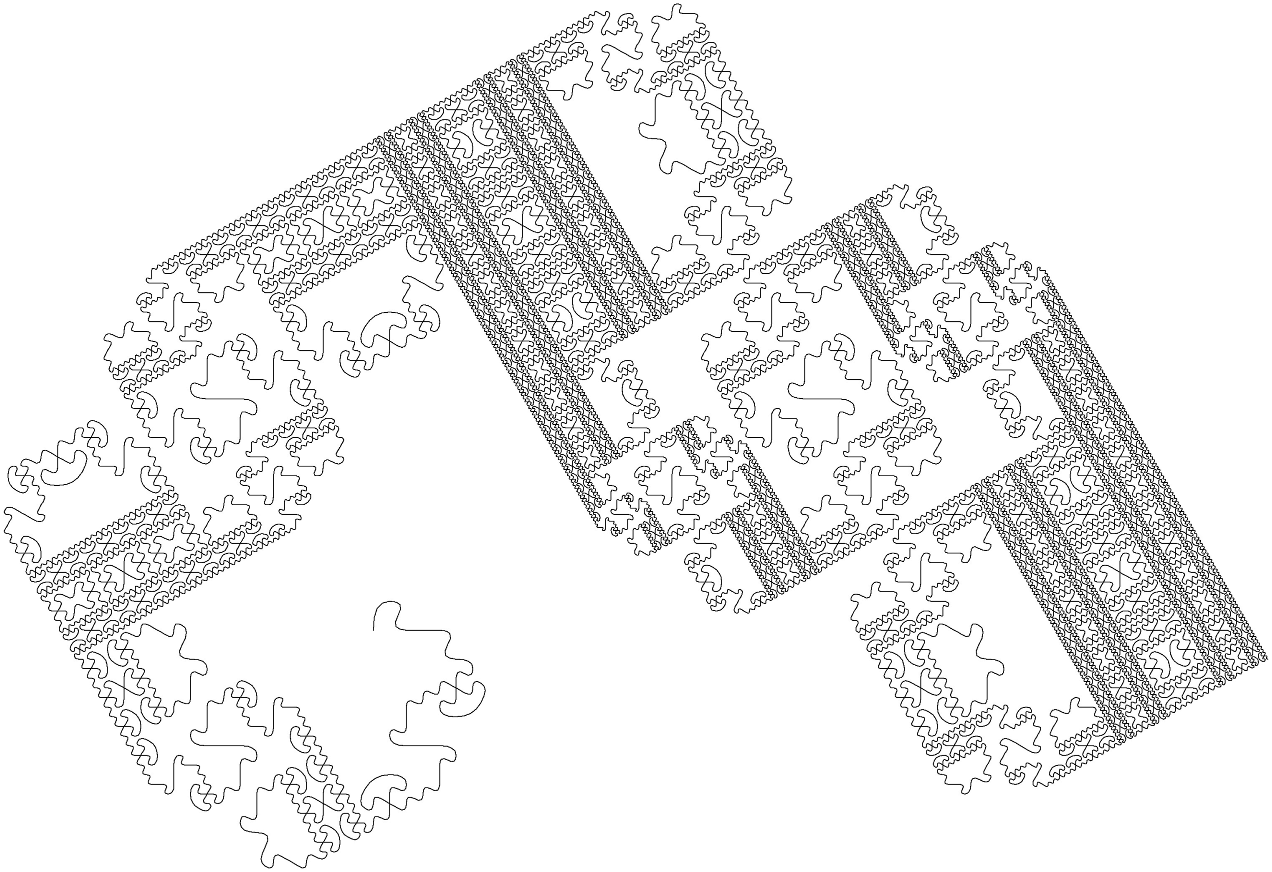


Here’s a variation on the same generator that makes another unraveled carpet. This one seems a bit more orderly in its unraveling.



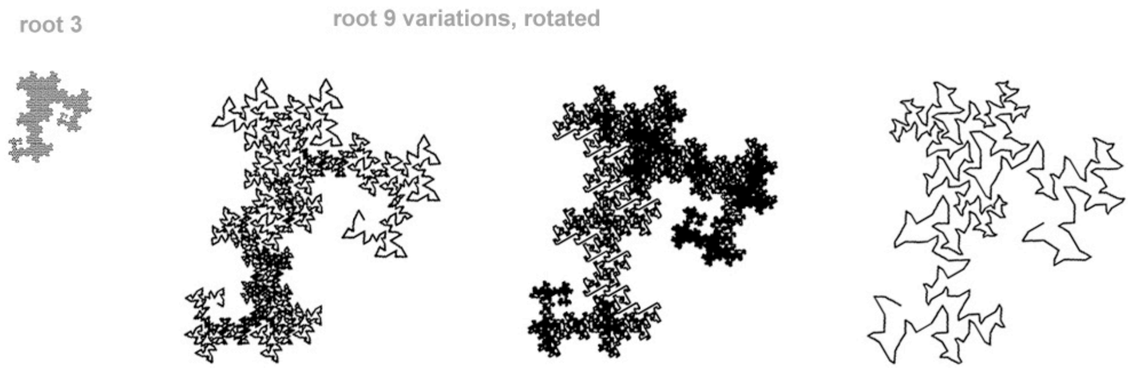
The specimen on the next page has unraveled itself in a strange way – and it is also a self-crosser.


interval length = √9
fractal dimension = 2.0
Triangular grid
7 segments
segment values:
1: –1, 2, 1, 1
2: 1, 0,–1,–1
3: 0,–1, 1,–1
4: 1, 0,–1,–1
5: 1, 0, 1, 1
6: 0,–1,–1, 1
7: 1, 0, 1, 1
generator
level 2
level 4 (rounded corners)

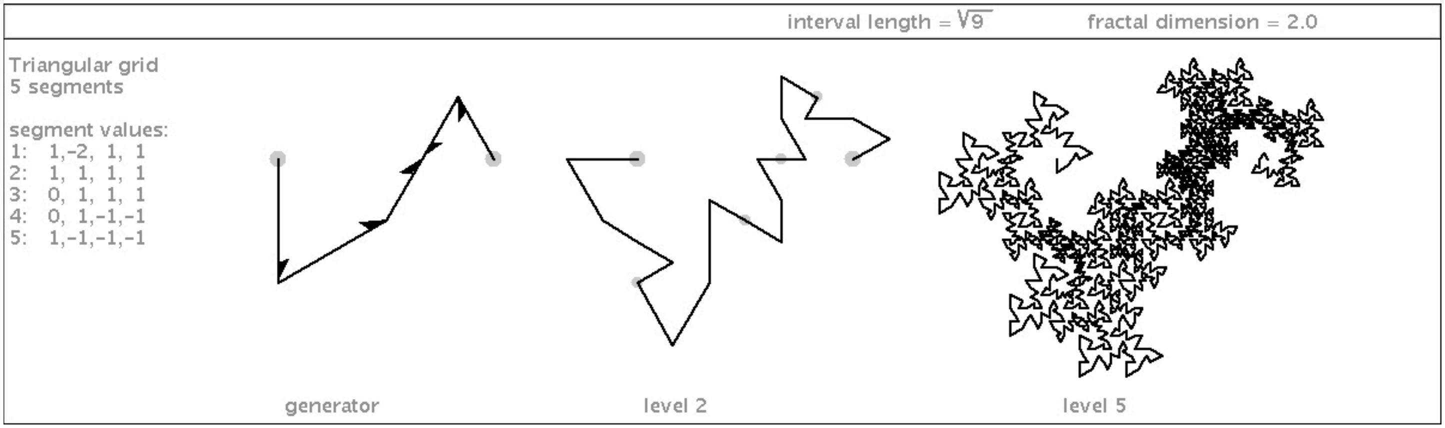


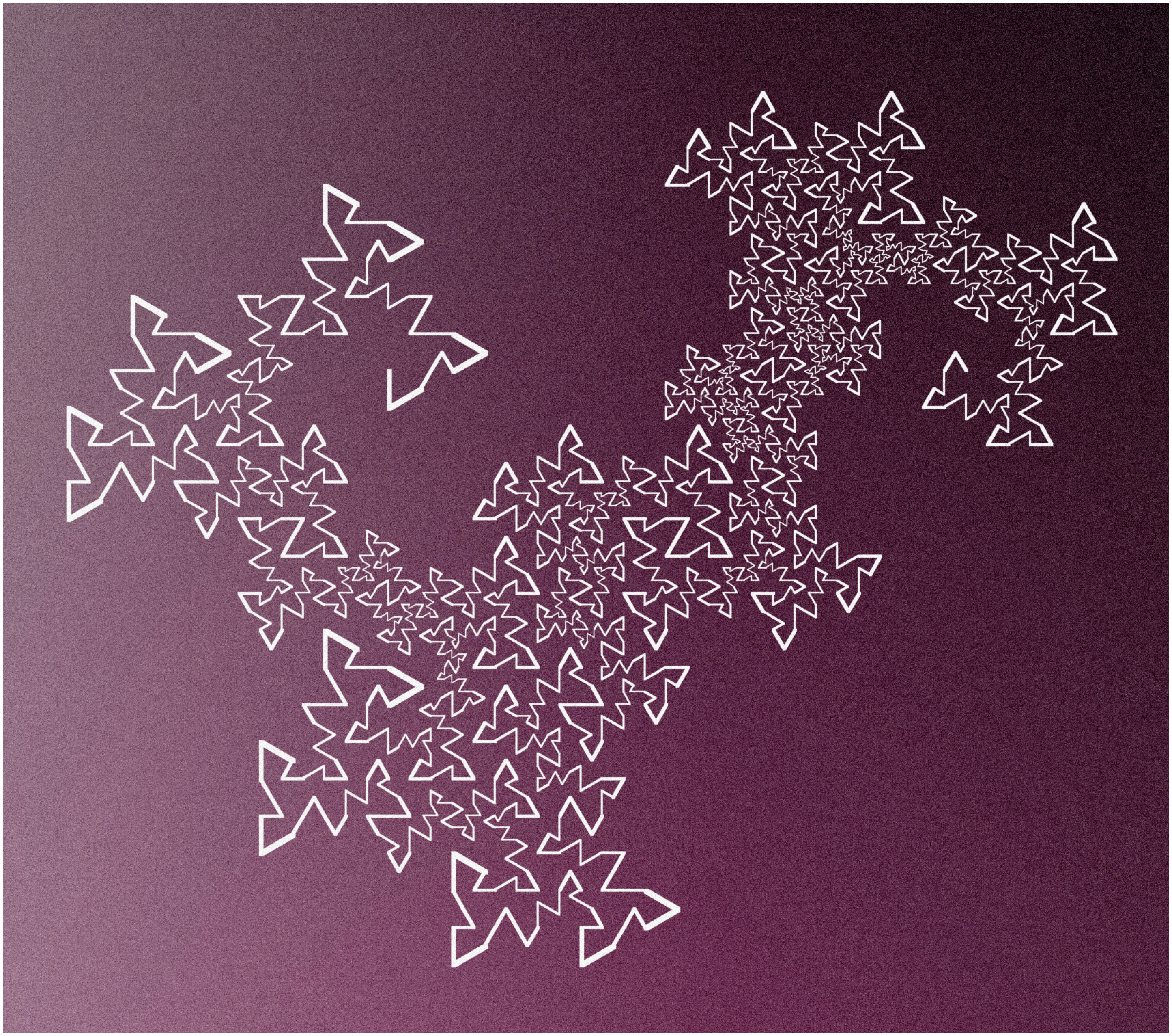
Where Be Dragons?

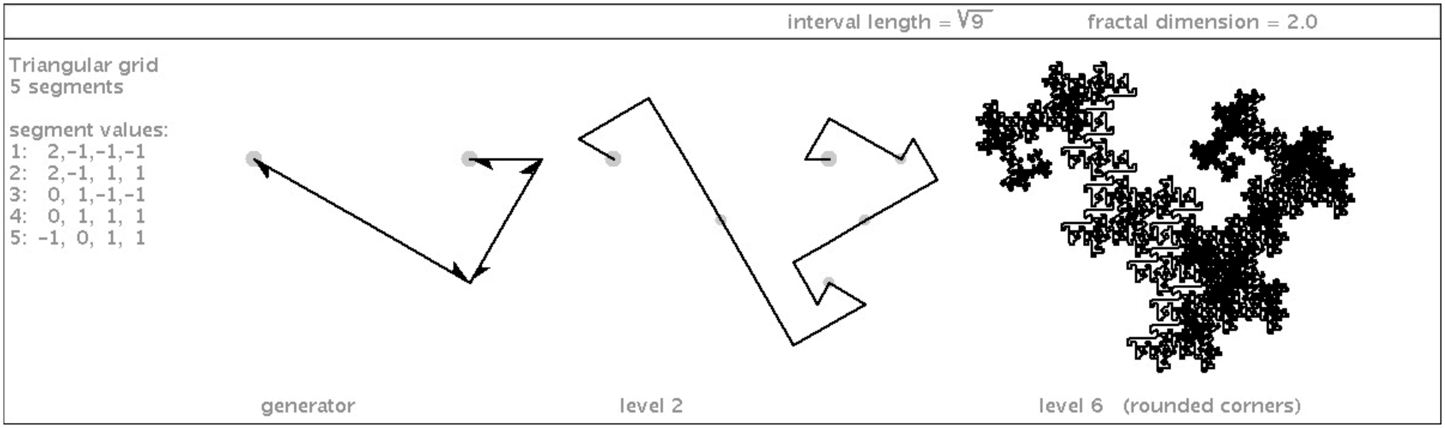
Are there any dragons in this family? In fact there is a whole clan of them. This clan of dragons consist of complexified 3x-scaled variations of a √3 specimen we met early on in the book – shown at left. I found three, but I have a suspicion that there are many more hiding in the deep √9 sea.

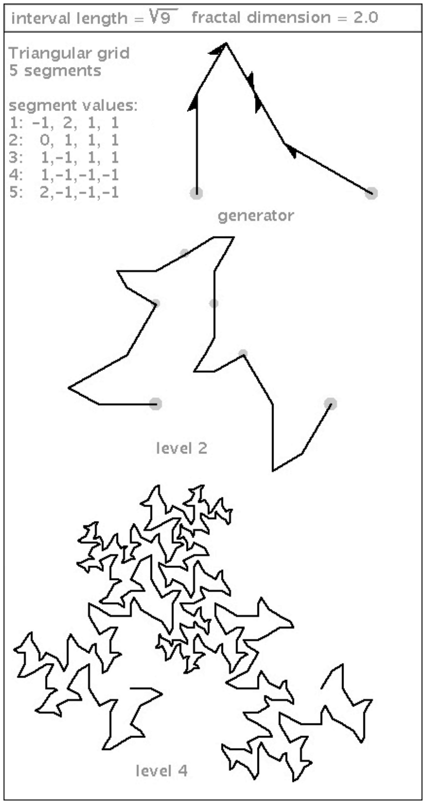


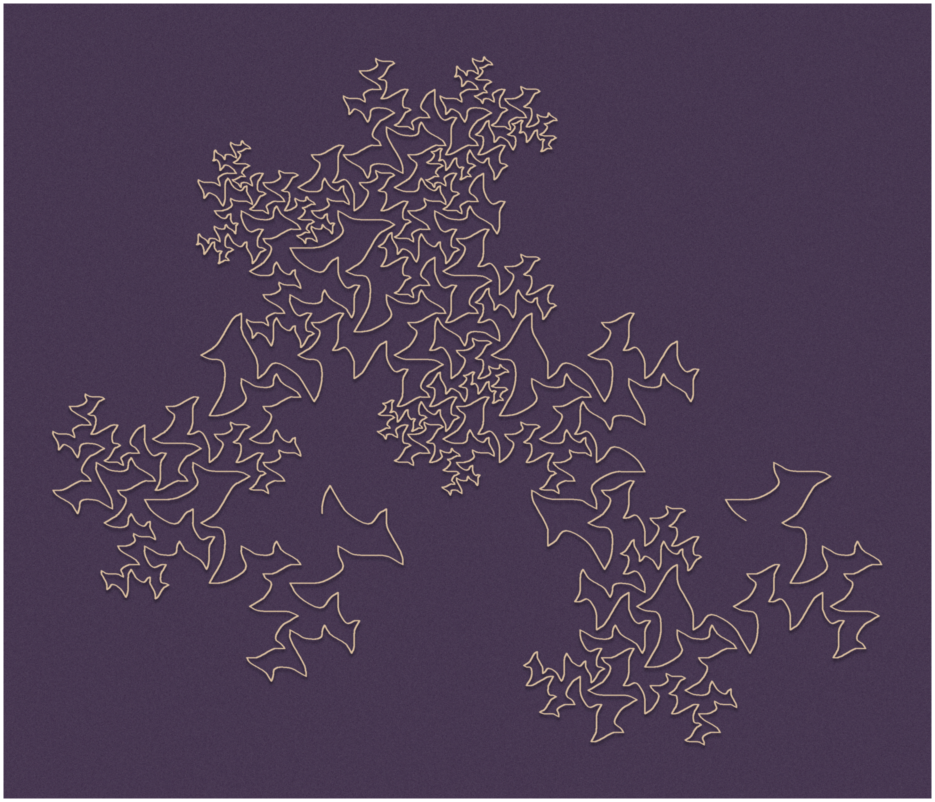
Each generator in this clan of dragons has five segments. Two of those segments have a length of √3, and the remaining three segments are of length 1. As I showed you earlier, squaring each of these lengths and then adding them results in a number that determines the fractal dimension. In this case, we get 9. On the next page is a color rendering of the self-avoiding curve below. And then I'll show you two others in the clan.



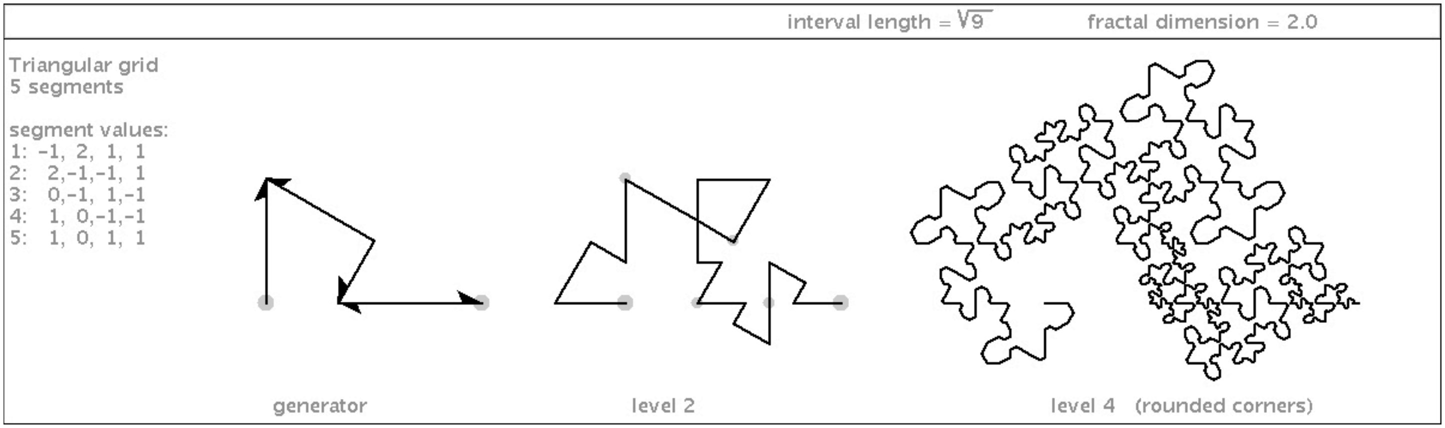



interval length = √9
fractal dimension = 2.0
Triangular grid
5 segments
segment values:
1: 2,-1,-1,-1
2: 2,-1, 1, 1
3: 0, 1,-1,-1
4: 0, 1, 1, 1
5: -1, 0, 1, 1
generator
level 2
level 6 (rounded corners)

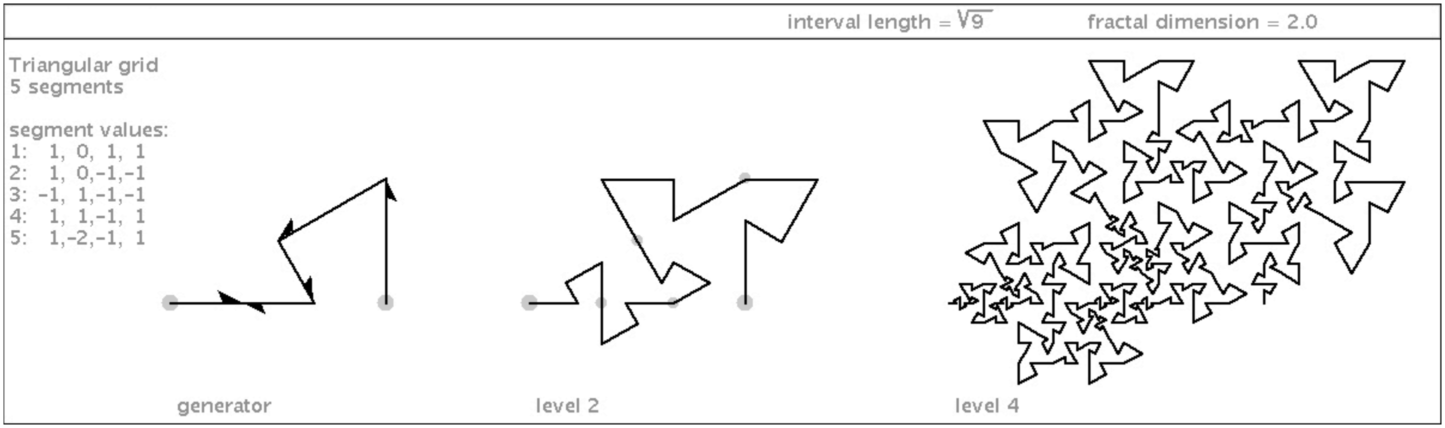

interval length = √9
fractal dimension = 2.0
Triangular grid
5 segments
segment values:
1: -1, 2, 1, 1
2: 0, 1, 1, 1
3: 1,-1, 1, 1
4: 1,-1,-1,-1
5: 2,-1,-1,-1
generator
level 2
level 4

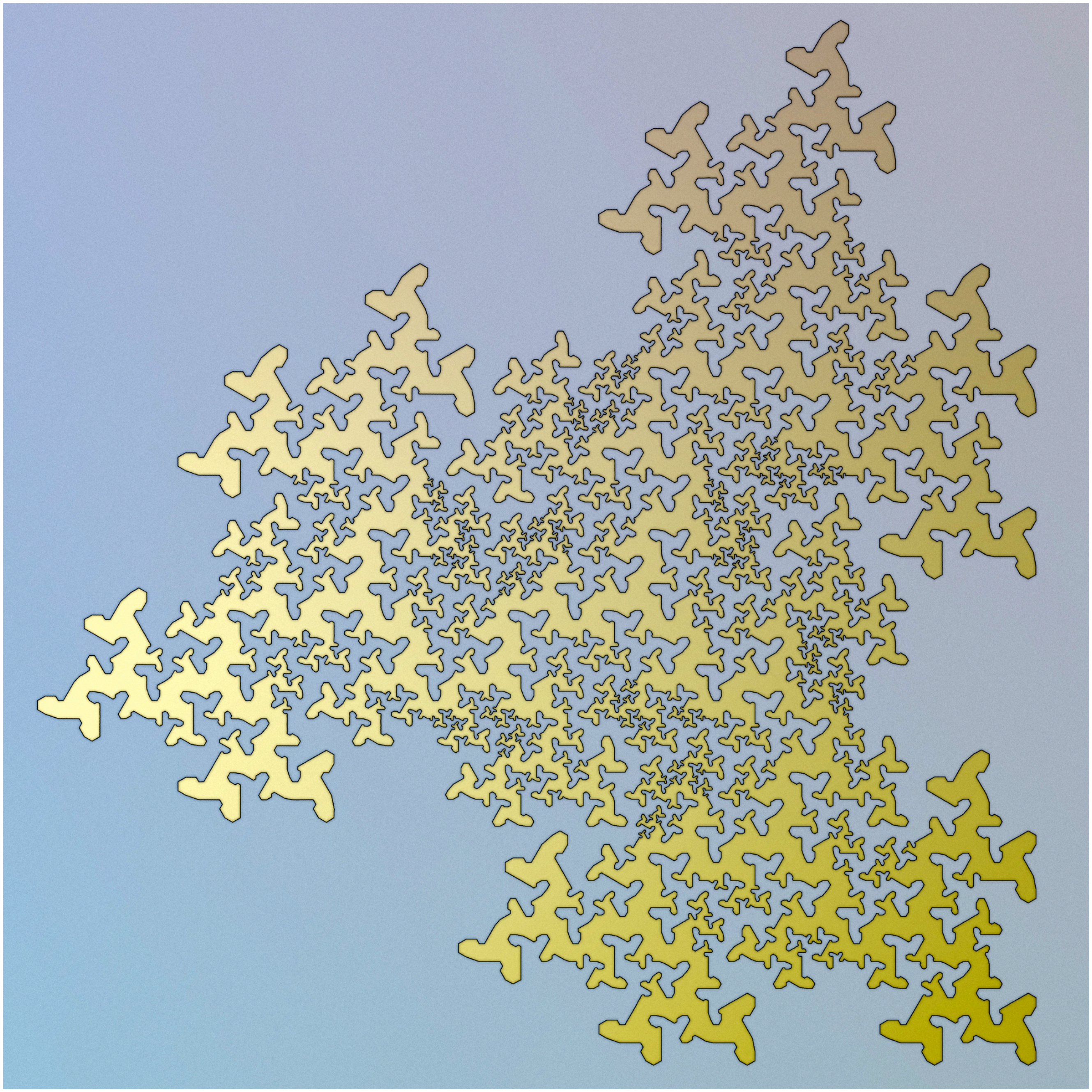


This specimen resolves to the same shape as the orderly unraveled carpet I just showed you.

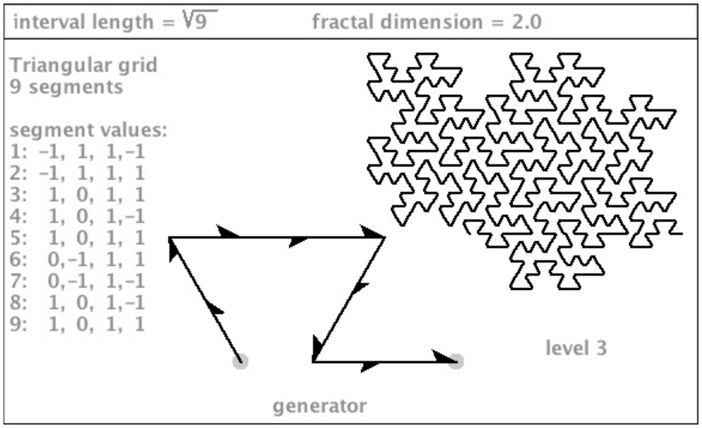


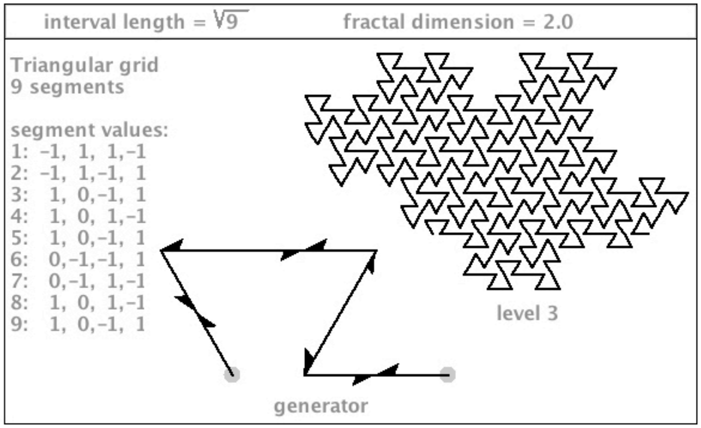
This next specimen is a self-avoider. On the next page I show it in color with rounded corners, on a triangle initiator.



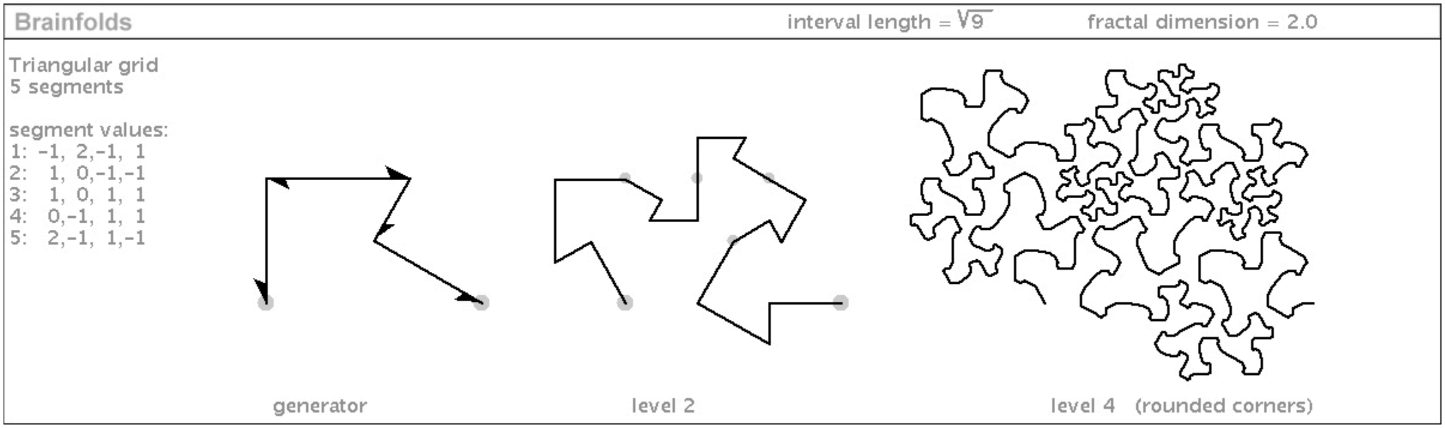


These two self-avoiders, based on the same generator, are quite similar, but look closely: you can see subtle differences.

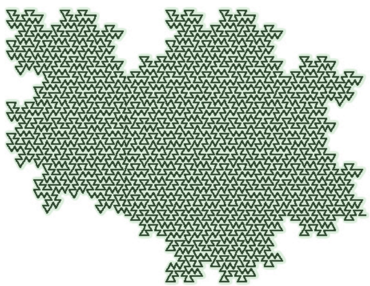


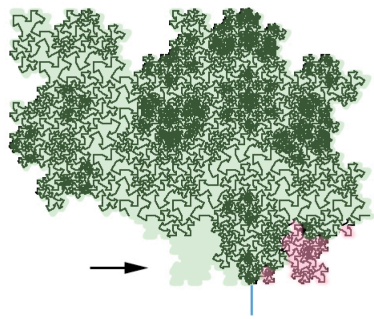


Here's another self-avoider. I call it "Brainfolds". (It was drawn by the turtle early on in the book).

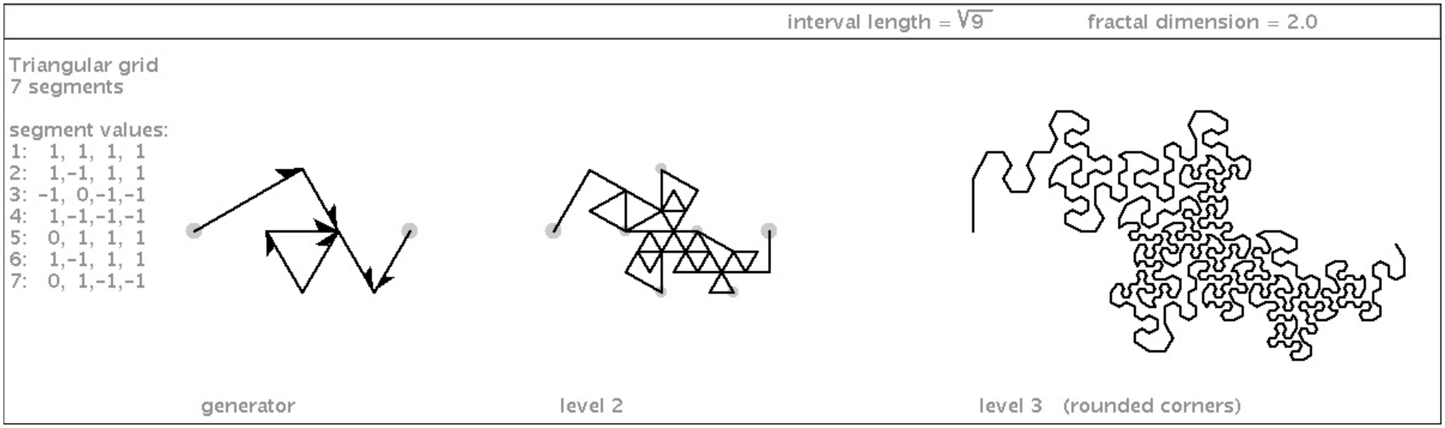


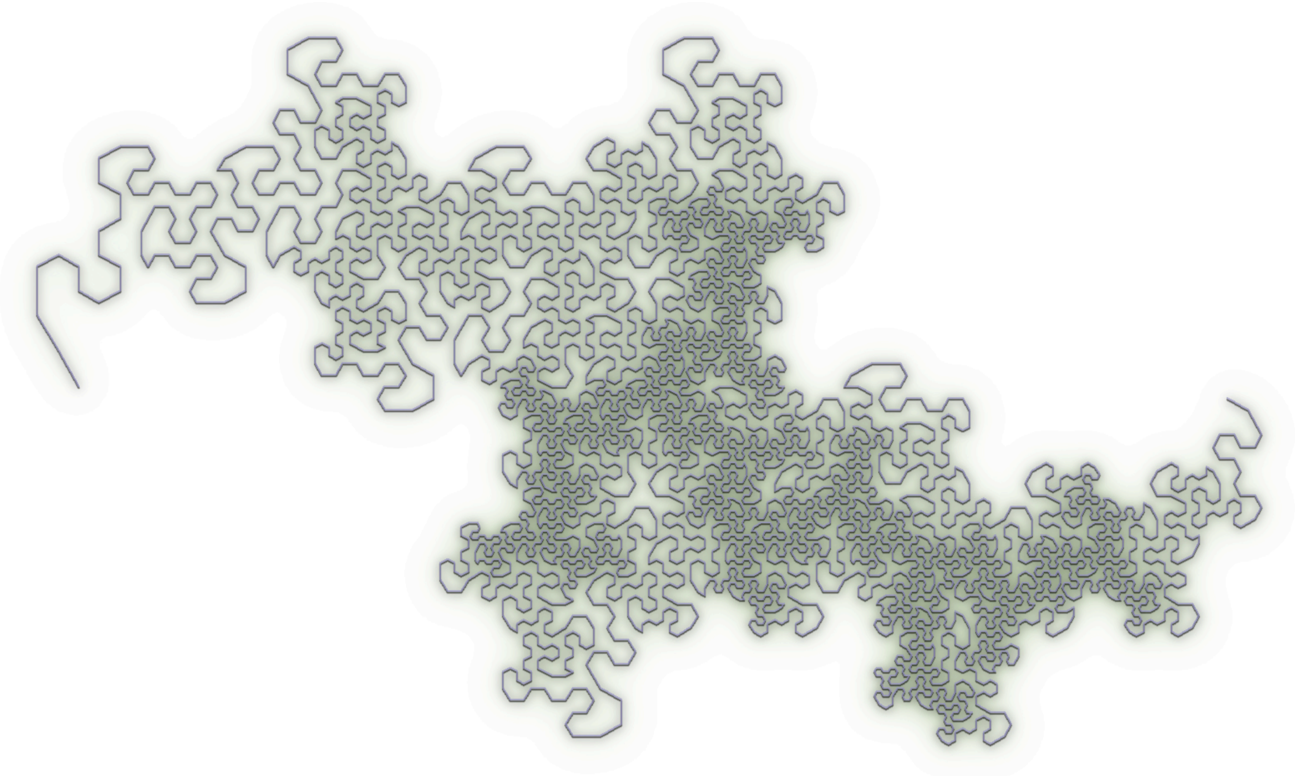
I had assumed that *Brainfolds* would fractalize to the same shape as the specimen at the upper-left of this page. My estimation was not quite right: there seems to be at least one small difference, shown at right: The pink region is a mirror image of the green region indicated by the arrow.



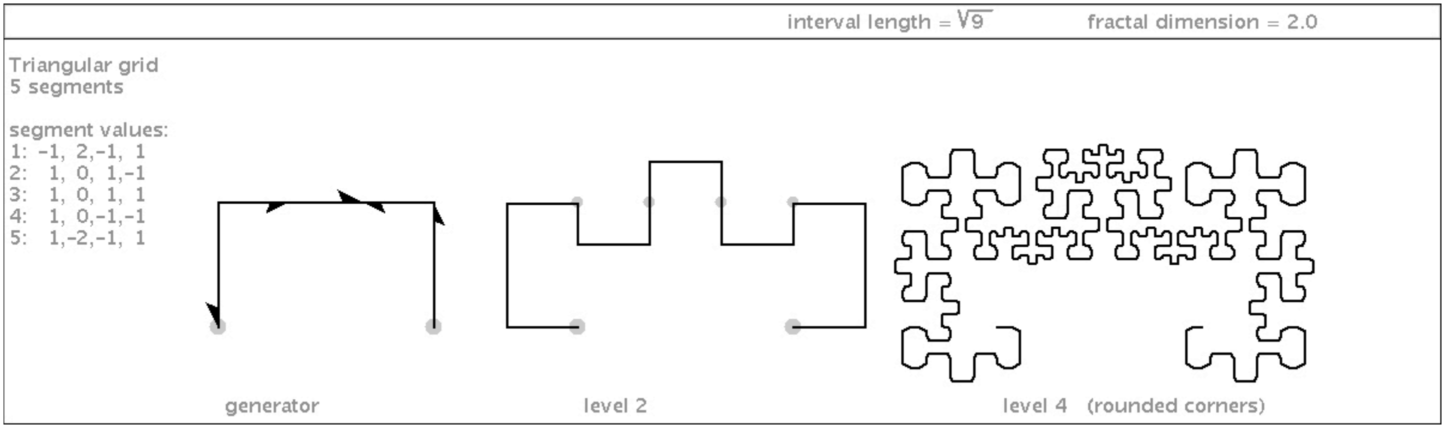


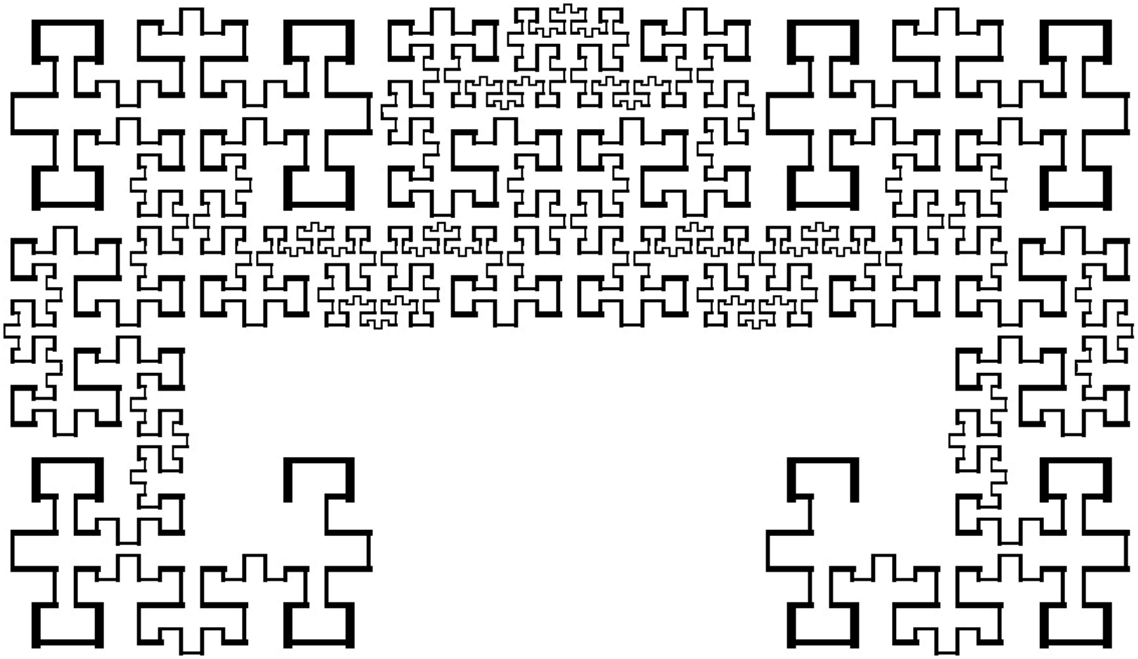
This next curve resolves to a 3x-size Ter-Dragon. Notice that the first segment in the generator has a length of √3, and that the rest of the generator consists of two Ter-Dragon generator-like shapes.



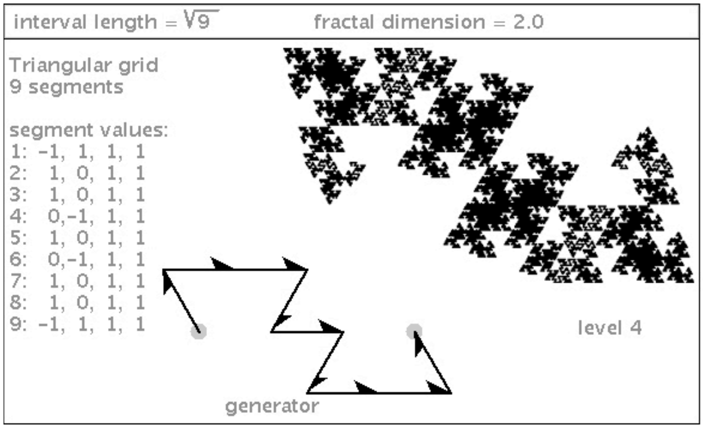


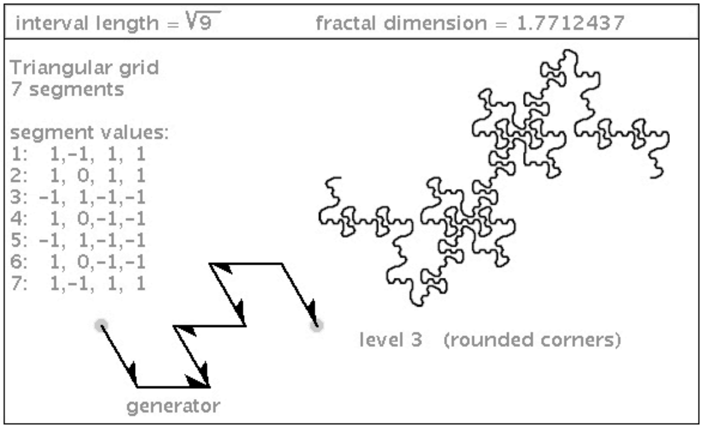
Here is a curve whose horizontal segments have length 1 and whose vertical segments have length √3. Even though it lives in the triangular grid, it has 90-degree angles. But that doesn't qualify it as a member of the square grid family! And check it out – this is a natural self-avoider.



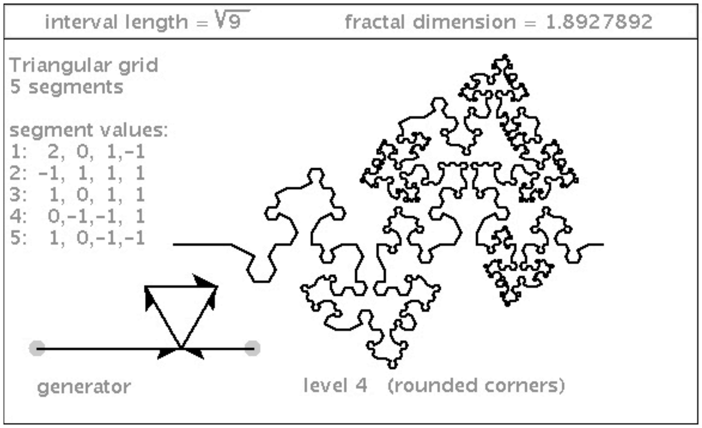


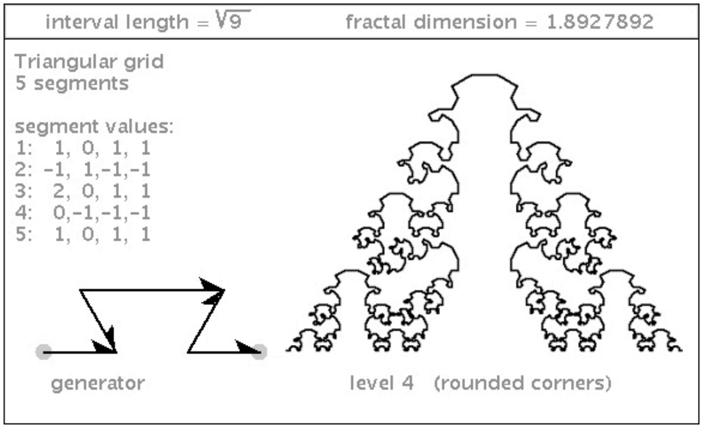
Here are a few rather zig-zaggy specimens: one is a busy dragon-like gridfiller and the other has a dimension of ~1.77.

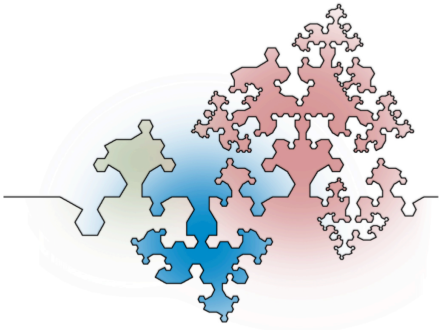


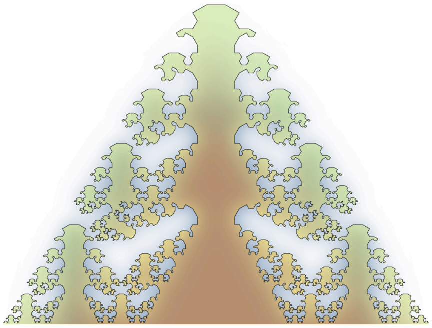


The remaining specimens of this family that I will show you have dimension ~1.89.

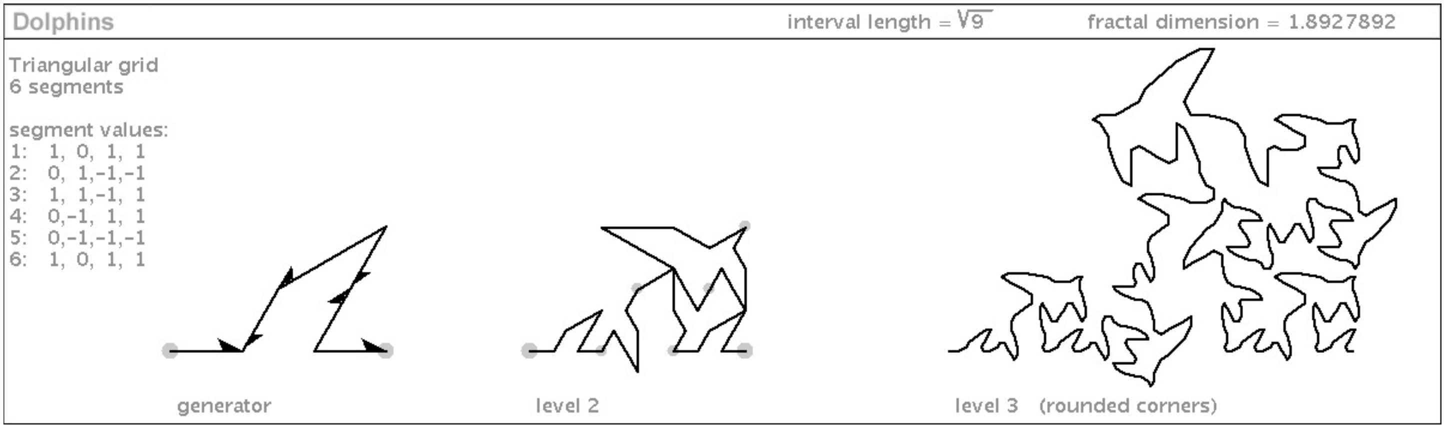




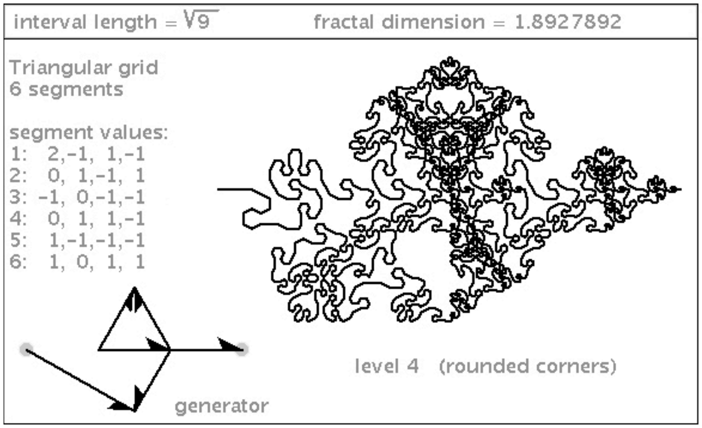


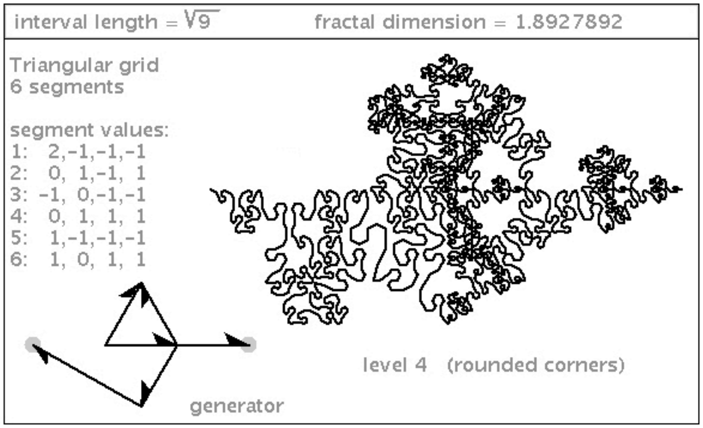


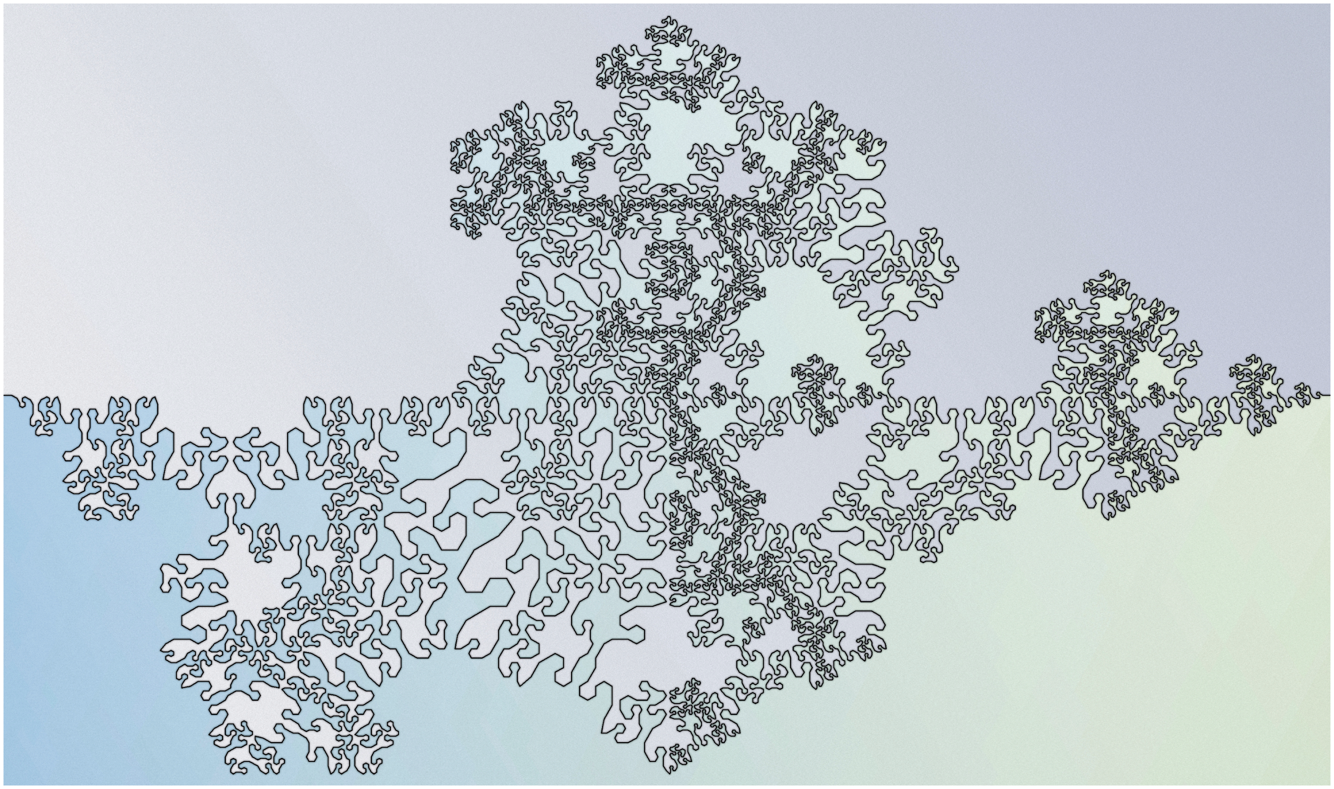
I call this one "Dolphins"



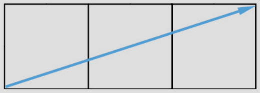
These last two curves are true gems. They are both based on the same generator shape. This first curve is shown at the very beginning of the book, and the second one is shown below.





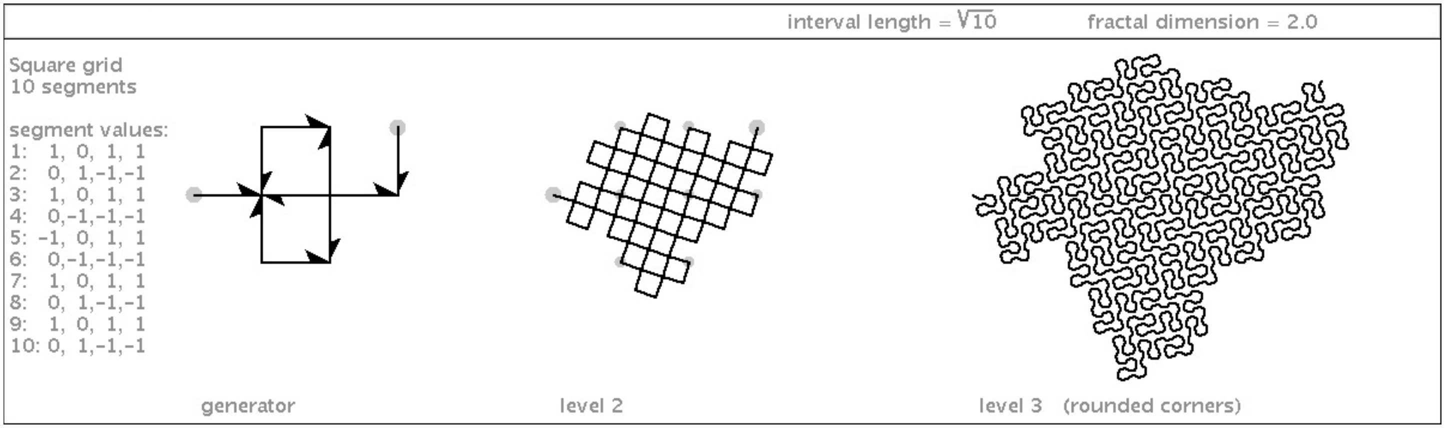


$\sqrt{10}$

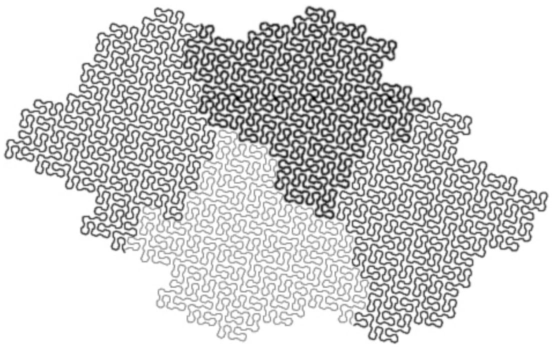


After coming down from seeing the amazing variety of the √9 triangle grid family, anything would seem anti-climactic. And indeed, I must report to you that the √10 family has very little to offer the eye and brain. Take a look and see for yourself. Why is this? I cannot say for sure, but I believe there must be some mathematically-relevant explanation.

I'll start with a rather non-descript specimen. It has a generator that looks like the classic Peano generator, but with a raised forearm on the right side. The distance from the origin to the elbow of the arm is √9, and the distance to the tip of the hand is √10. By taking the Peano generator and adding the waving arm, and some alternate flippings, we get the following shape:



In an attempt to add some excitement, I will pertile this specimen four times. Here it is shown at right.

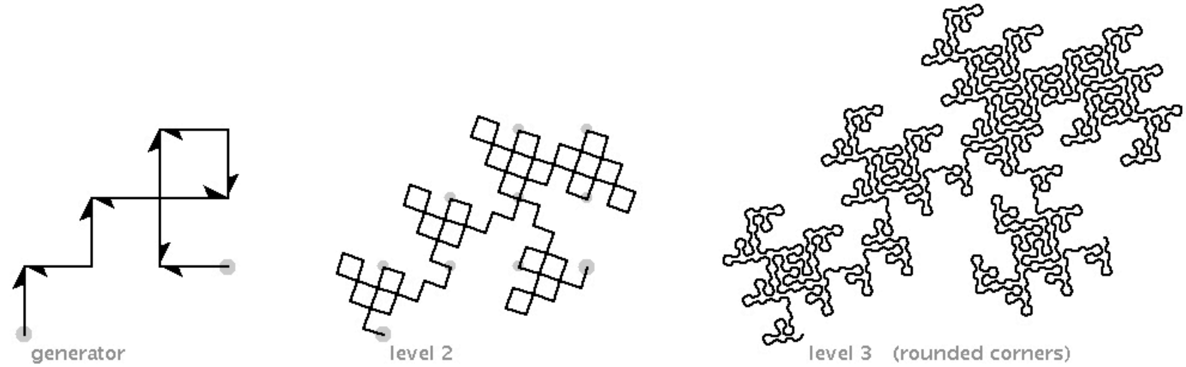


Notice the alternating pattern in the flippings of the genetic code. I found that alternating the flippings helped me to locate several gridfillers of this family. The next four specimens use this alternating trick. On the following page are three specimens. They are all gridfillers; two of them are based on a common generator shape.

interval length = $\sqrt{10}$ fractal dimension = 2.0

Square grid
10 segments

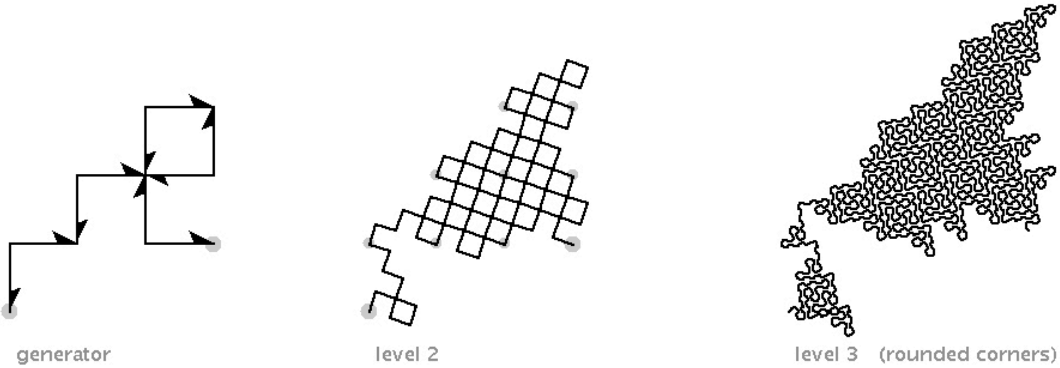
segment values:
1: 0, 1, 1, 1
2: 1, 0,-1,-1
3: 0, 1, 1, 1
4: 1, 0,-1,-1
5: 0, 1, 1, 1
6: 1, 0,-1,-1
7: 0,-1, 1, 1
8: -1, 0,-1,-1
9: 0,-1, 1, 1
10: 1, 0,-1,-1



interval length = $\sqrt{10}$ fractal dimension = 2.0

Square grid
10 segments

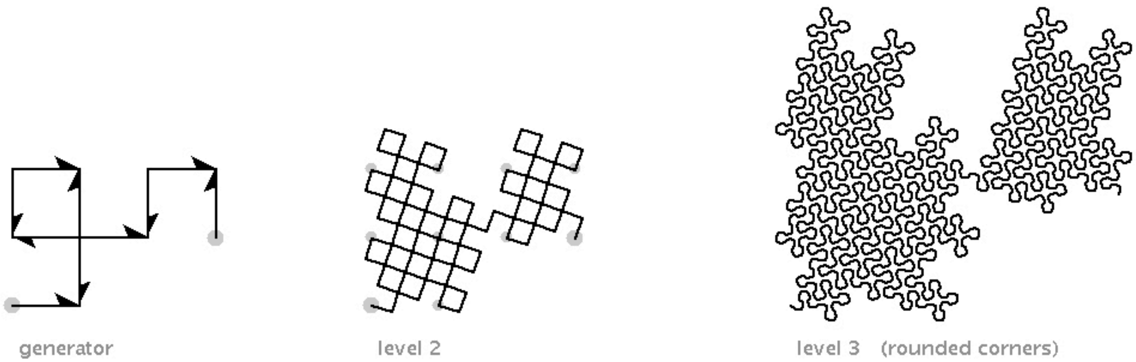
segment values:
1: 0, 1,-1,-1
2: 1, 0, 1, 1
3: 0, 1,-1,-1
4: 1, 0, 1, 1
5: 0, 1,-1,-1
6: 1, 0, 1, 1
7: 0,-1,-1,-1
8: -1, 0, 1, 1
9: 0,-1,-1,-1
10: 1, 0, 1, 1



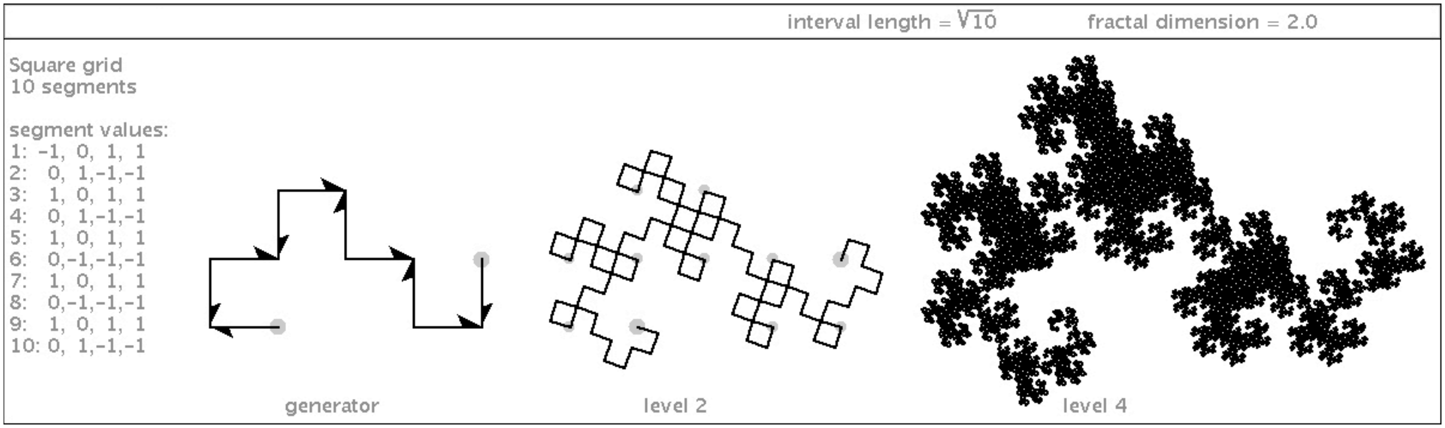
interval length = $\sqrt{10}$ fractal dimension = 2.0

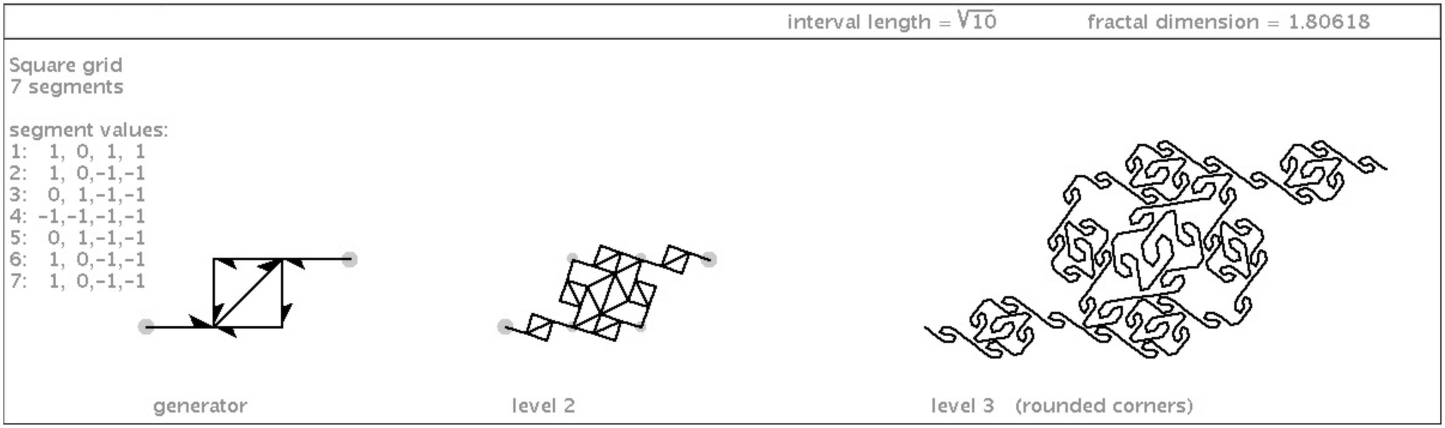
Square grid
10 segments

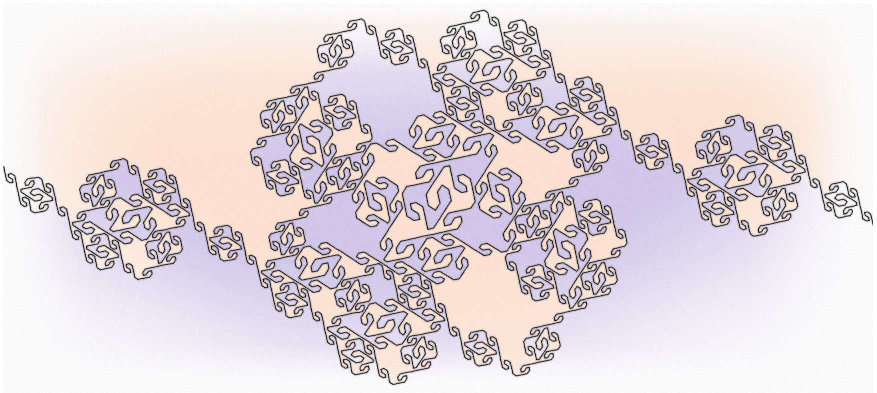
segment values:
1: 1, 0, 1, 1
2: 0, 1,-1,-1
3: -1, 0, 1, 1
4: 0, 1,-1,-1
5: 1, 0, 1, 1
6: 0,-1,-1,-1
7: 1, 0, 1, 1
8: 0, 1,-1,-1
9: 1, 0, 1, 1
10: 0,-1,-1,-1



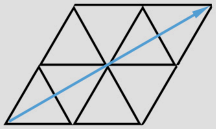
The last two curves I will show of the √10 family is a dragon and a palindrome with dimension ~1.806.



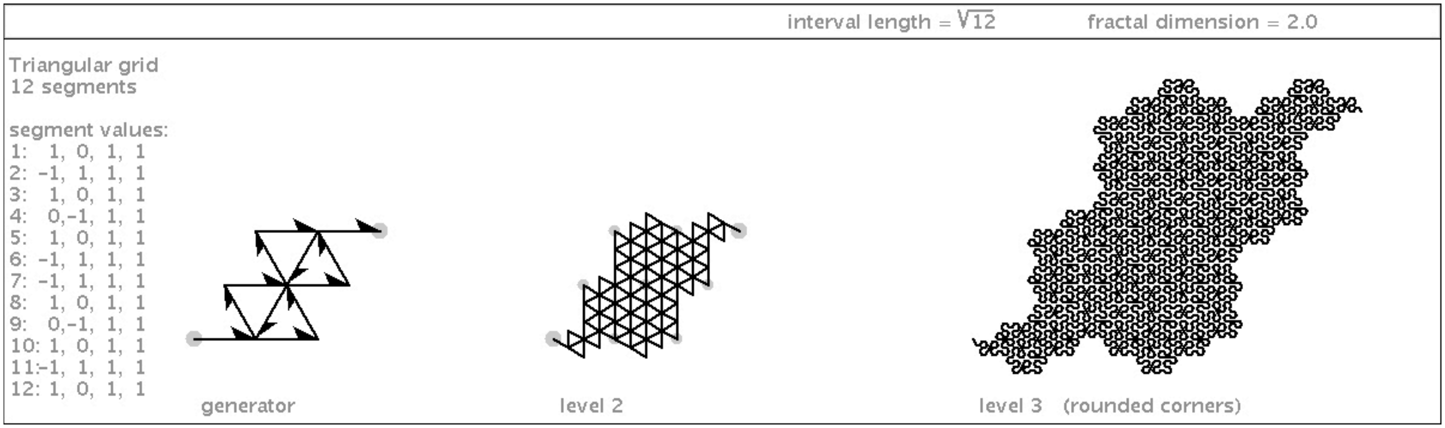




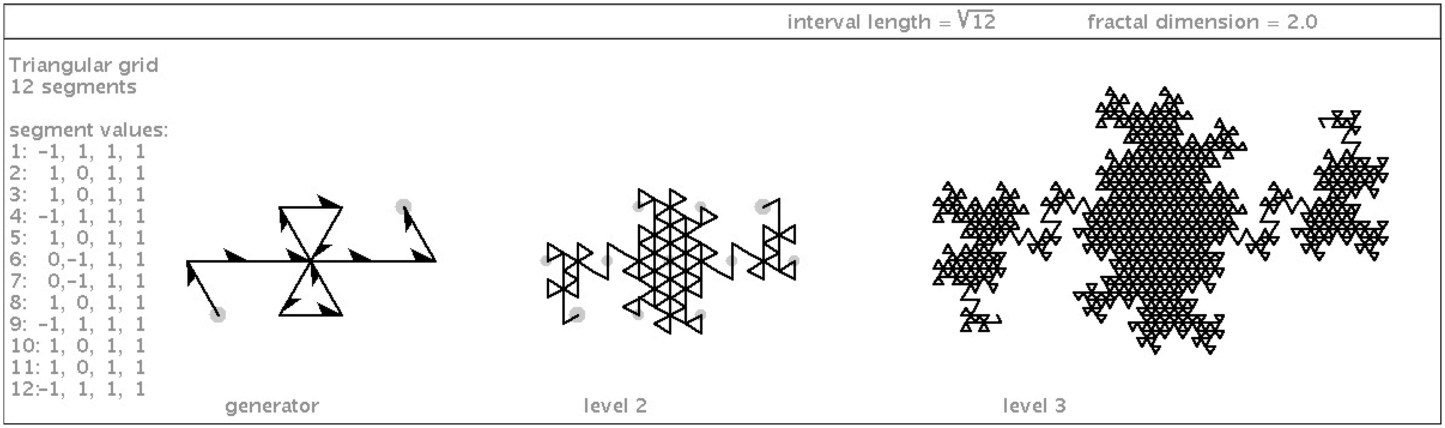
√12



It is time to show you another overweight palindrome dragon. Like many of the others we've seen, this member of the √12 family has a typical *yam-like* shape: fat and lumpy in the middle; tapered at the ends.



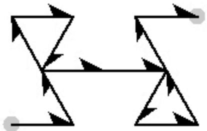
Given the many ways you can traverse a triangular grid in the interval of √12, we should expect many more palindrome dragons. I'll show you four more. These are not at all yam-like. In fact, their boundaries are quite craggy.

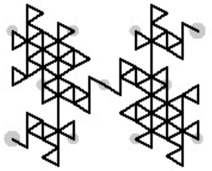


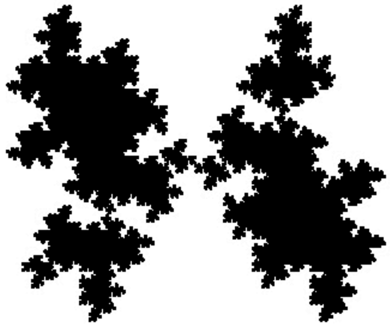
interval length = $\sqrt{12}$ fractal dimension = 2.0

Triangular grid
12 segments

segment values:
1: 1, 0, 1, 1
2: -1, 1, 1, 1
3: -1, 1, 1, 1
4: 1, 0, 1, 1
5: 0,-1, 1, 1
6: 1, 0, 1, 1
7: 1, 0, 1, 1
8: 0,-1, 1, 1
9: 1, 0, 1, 1
10:-1, 1, 1, 1
11:-1, 1, 1, 1
12: 1, 0, 1, 1


generator

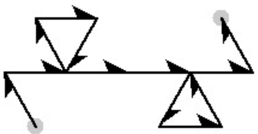

level 2

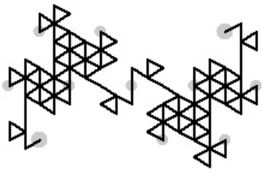

level 4 (rounded corners)

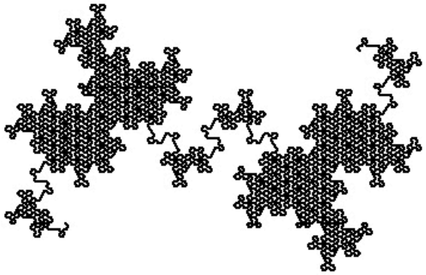
interval length = $\sqrt{12}$ fractal dimension = 2.0

Triangular grid
12 segments

segment values:
1: -1, 1, 1, 1
2: 1, 0, 1, 1
3: -1, 1, 1, 1
4: 1, 0, 1, 1
5: 0,-1, 1, 1
6: 1, 0, 1, 1
7: 1, 0, 1, 1
8: 0,-1, 1, 1
9: 1, 0, 1, 1
10:-1, 1, 1, 1
11: 1, 0, 1, 1
12:-1, 1, 1, 1


generator


level 2

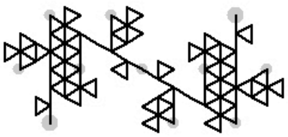

level 3 (rounded corners)

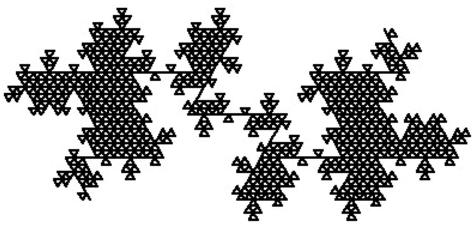
interval length = $\sqrt{12}$ fractal dimension = 2.0

Triangular grid
12 segments

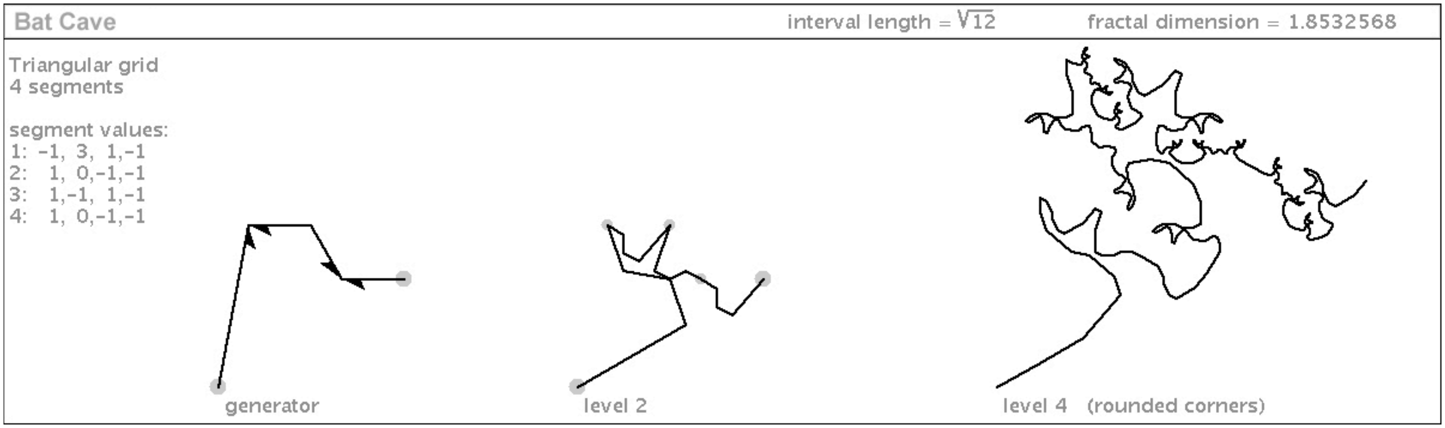
segment values:
1: 0, 1, 1, 1
2: -1, 0, 1, 1
3: 0, 1, 1, 1
4: 1,-1, 1, 1
5: 0, 1, 1, 1
6: 1,-1, 1, 1
7: 1,-1, 1, 1
8: 0, 1, 1, 1
9: 1,-1, 1, 1
10: 0, 1, 1, 1
11:-1, 0, 1, 1
12: 0, 1, 1, 1

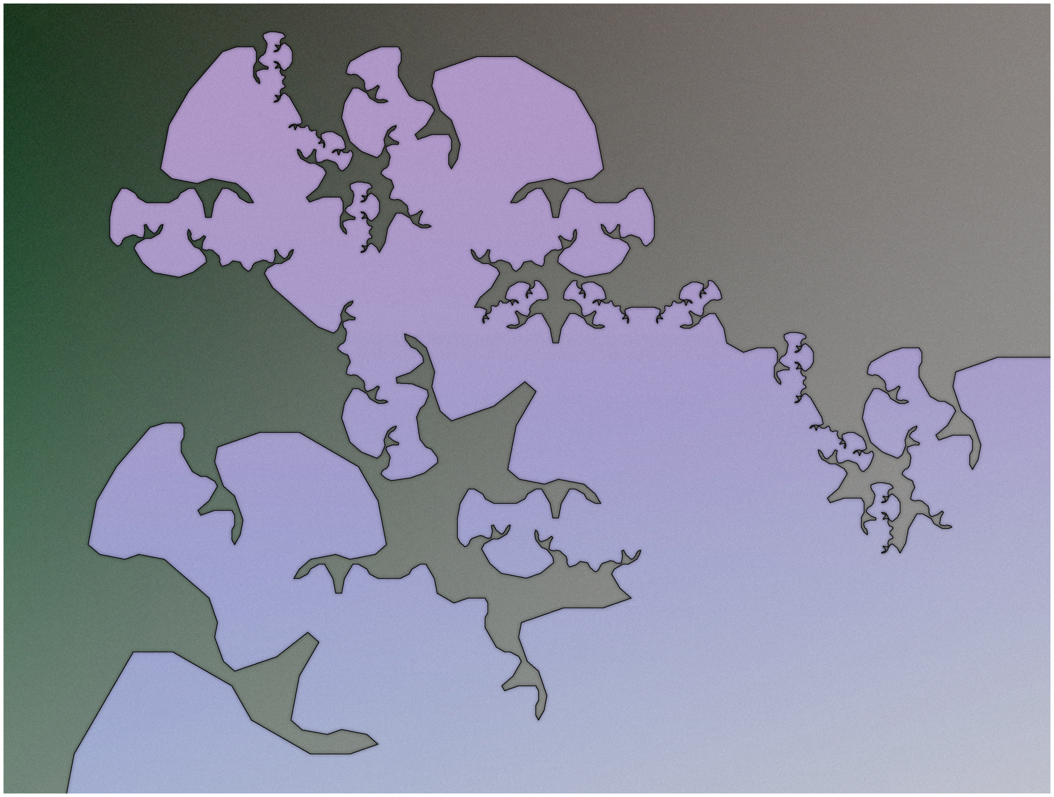

generator


level 2

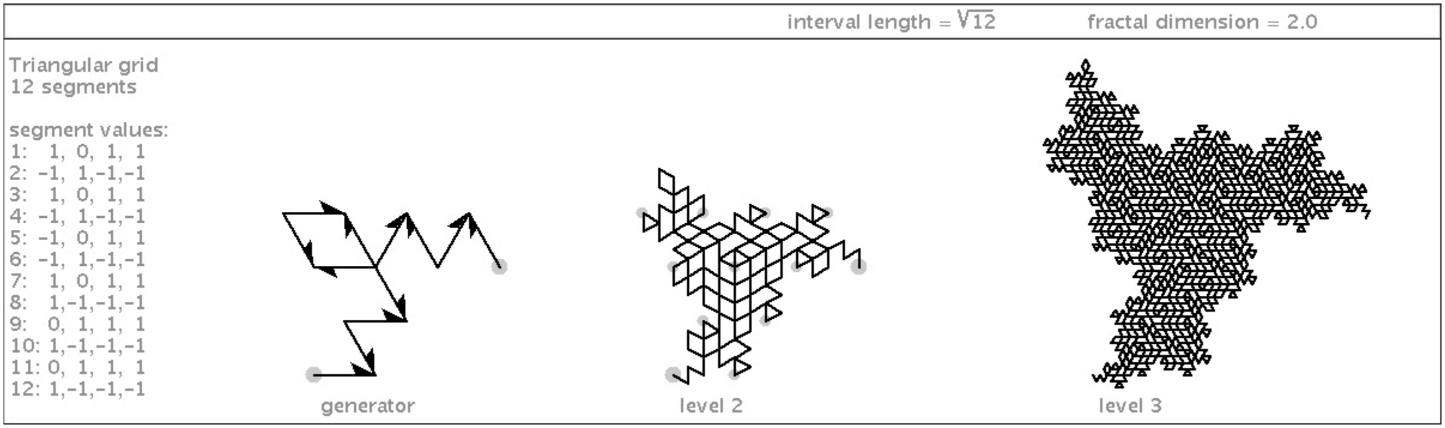

level 3

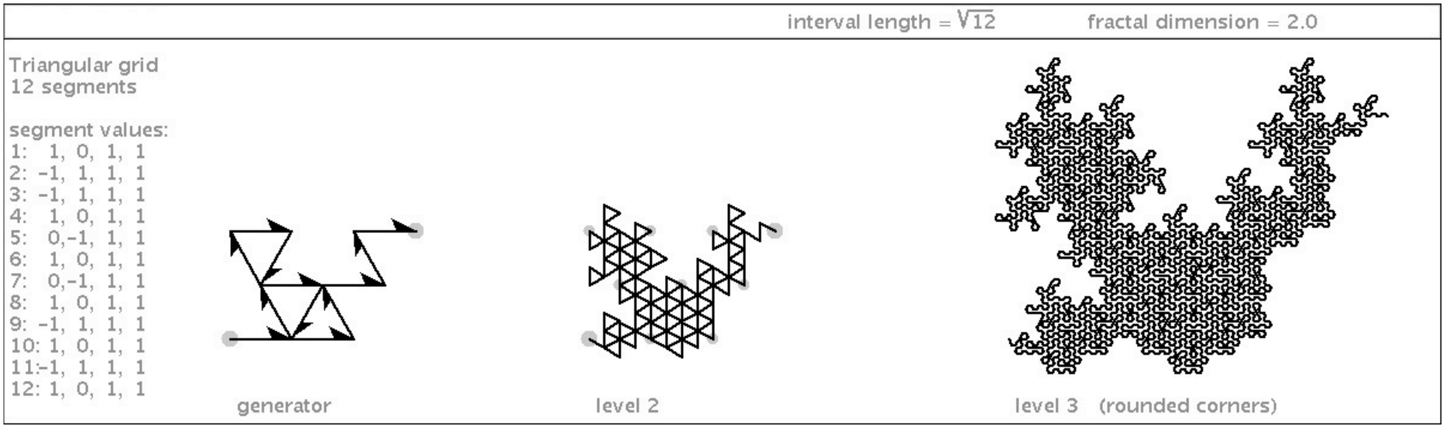
Here is a √12 specimen that includes an abnormally long segment – extending the length of √7, followed by three segments of length 1. Who would have ever expected that its 5th teragon would resemble a bat's cave?

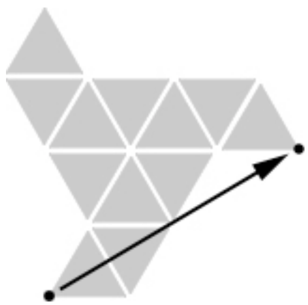


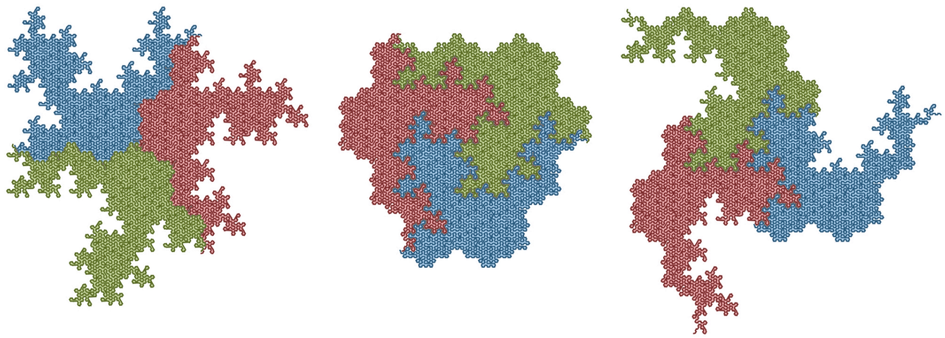


Here are two interesting specimens. The first one is based on a tiling of 12 triangles – illustrated at bottom-left. The second one is shown with two kinds of 3-way pertilings - shown at bottom-right.

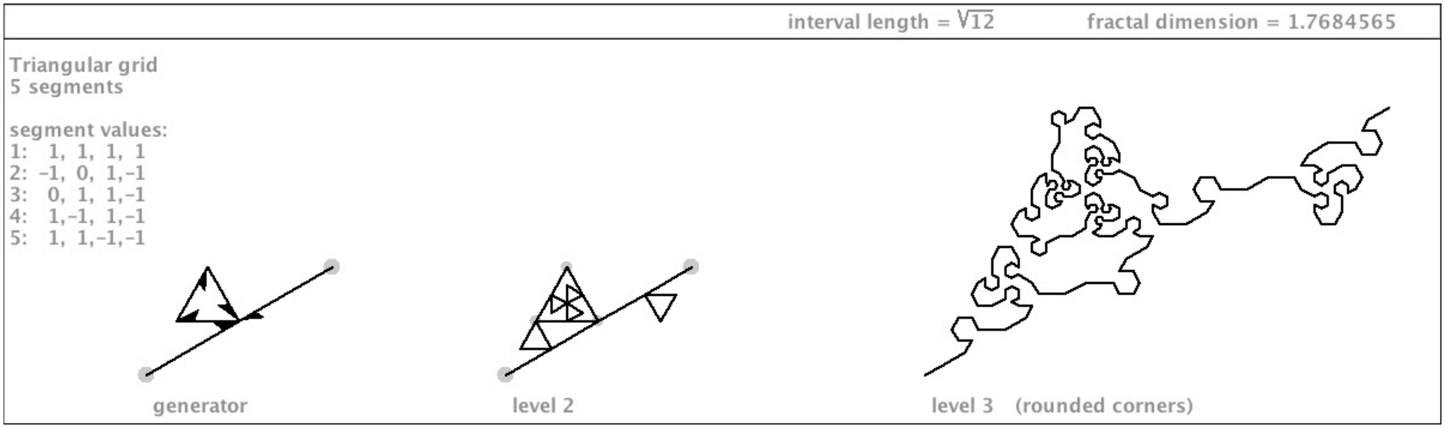


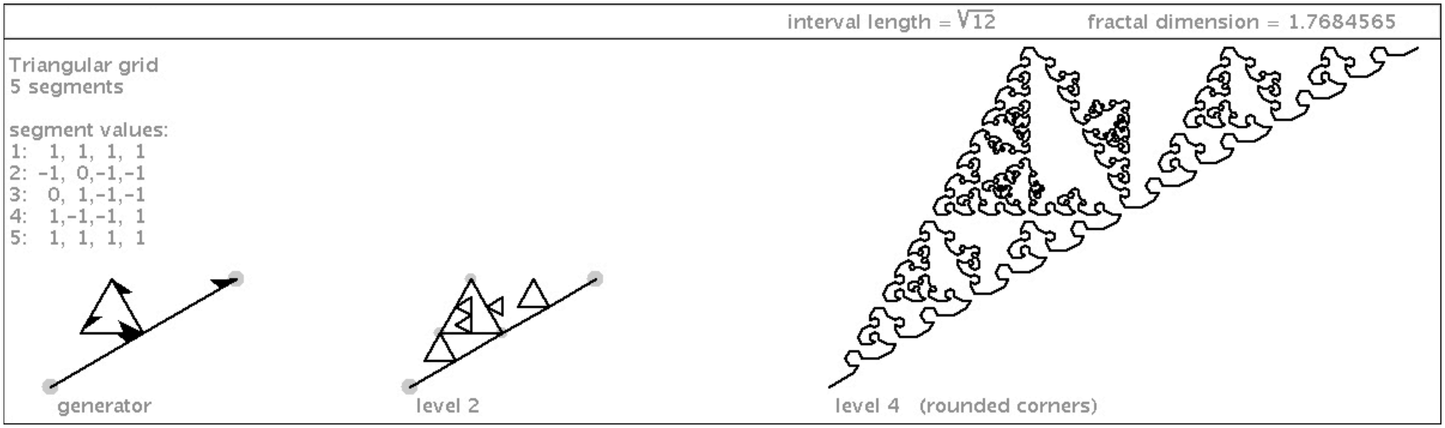


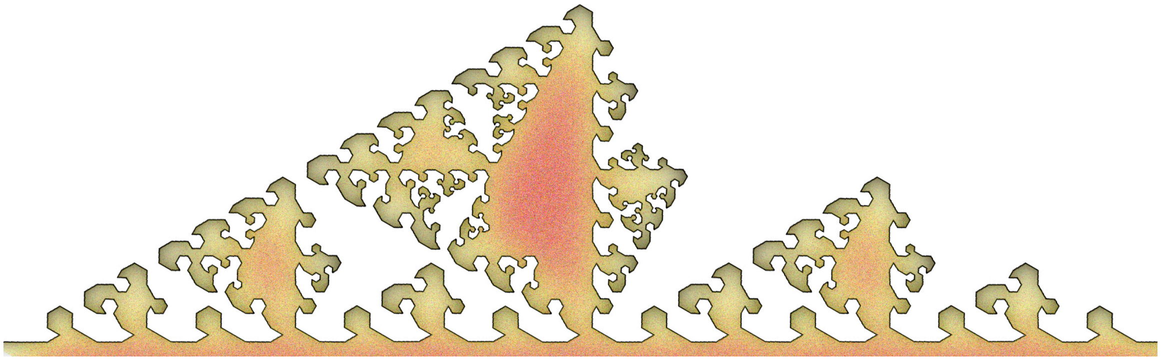




Here are a couple of curves based on a common generator – it includes two segments of length √3:

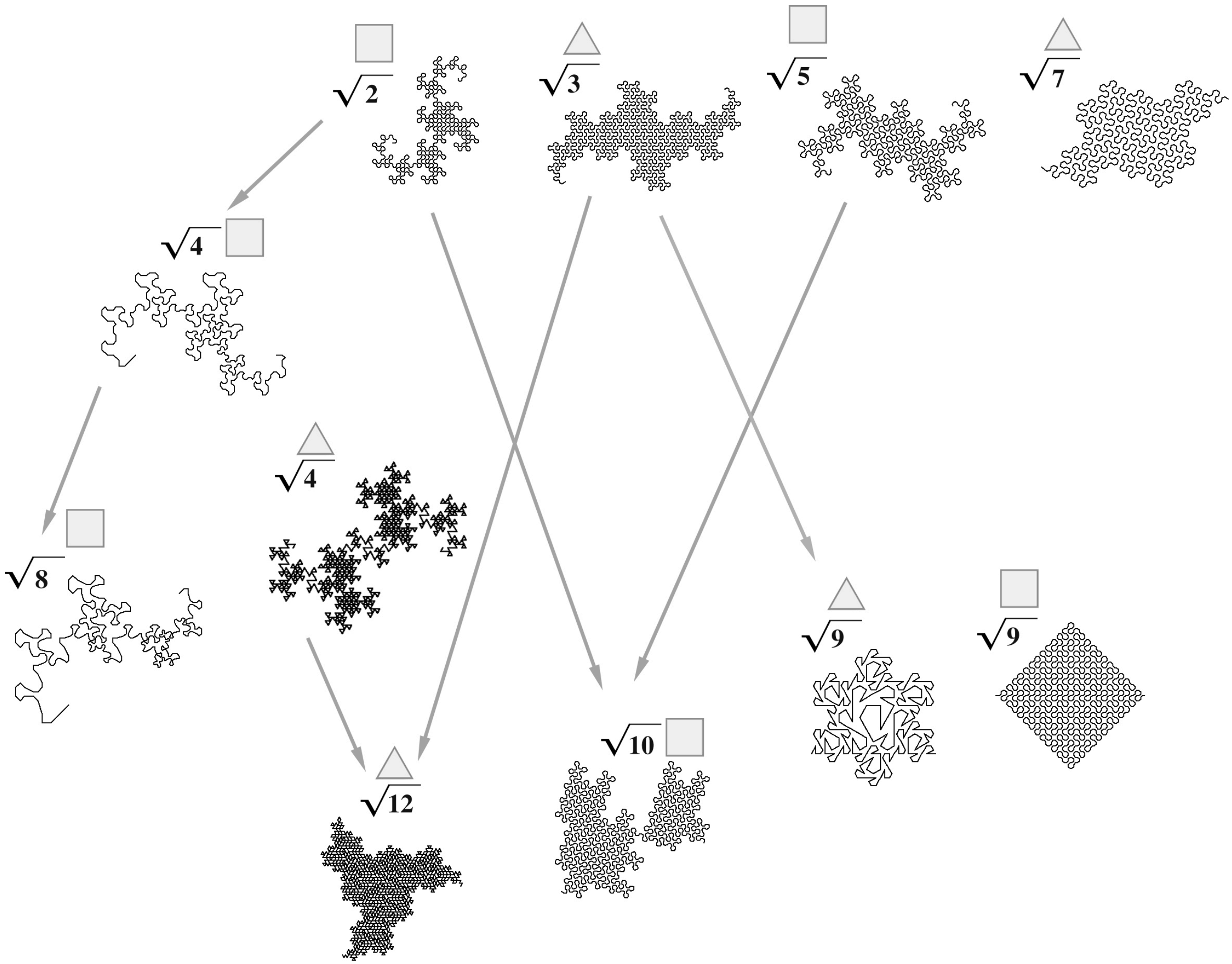




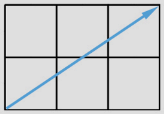


Ancestry

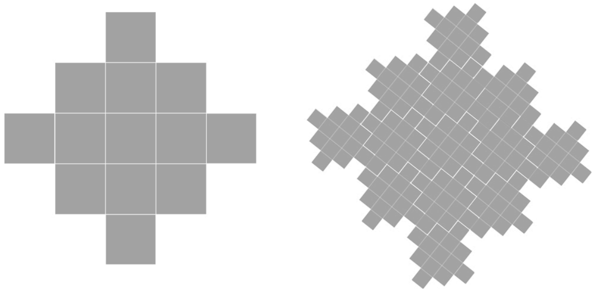
We have met several families of fractal curves, and we have seen a number of ways in which they relate to each other. In the graph below, I show examples from all families up to √12. The prime-numbered families are placed along the top. The powers-of-two families (2, 4, 8) are shown at left. Notice that variations of the HH Dragon can be generated within each of the power-of-two families. Similarly, any √3 curve can be generated within in the √9 family, since 9 is a multiple of 3. The √10 family can generate variations of both the √2 and √5 family curves. And finally, the √12 family can generate variations of √3 and √4 (triangle grid) curves.

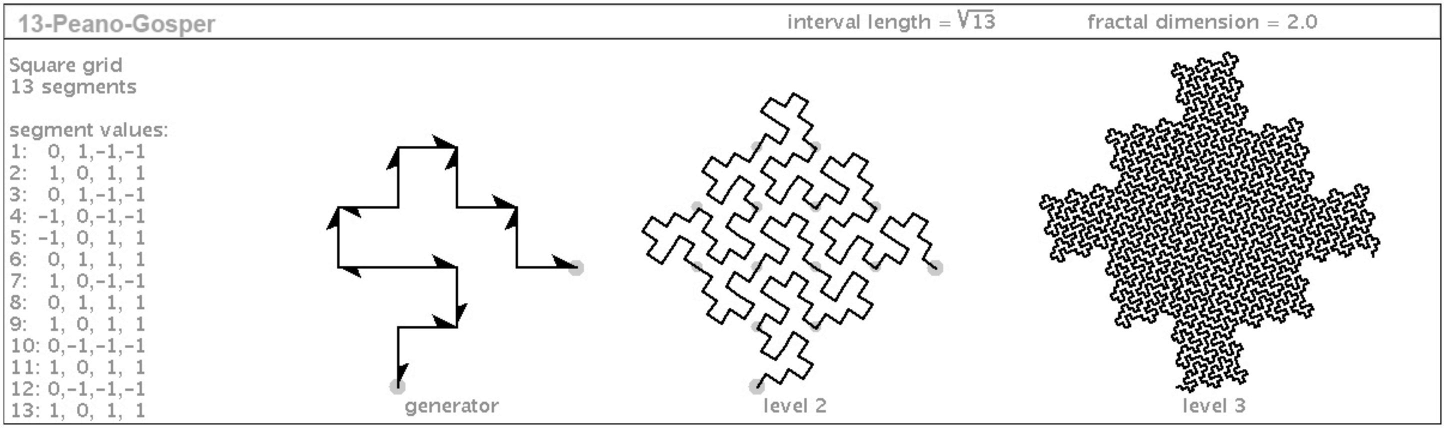


$\sqrt{13}$

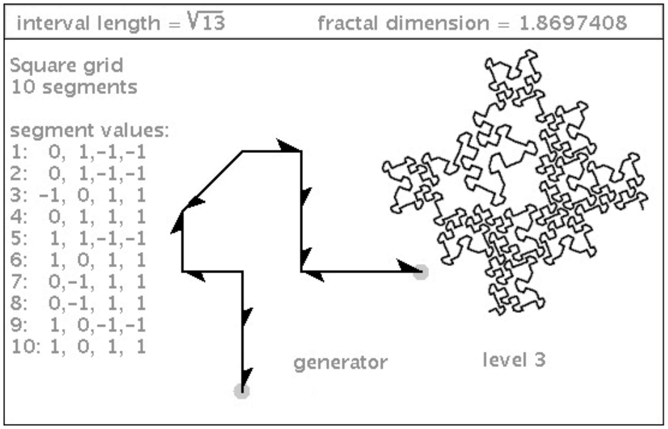


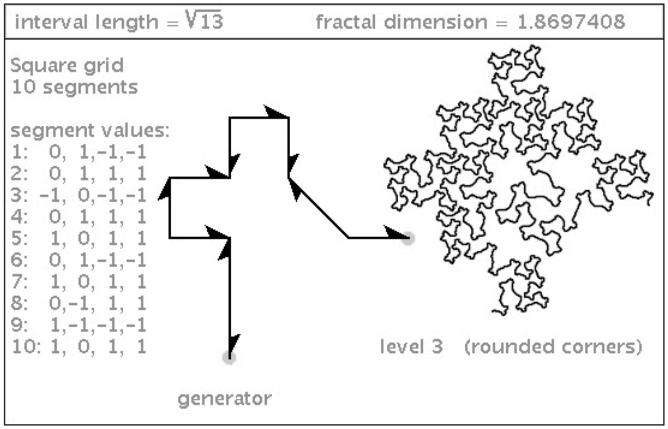
Now it is time to look at the √13 square grid family. I will start with a curve I discovered that is a self-avoider analogous to the Gosper curve: it requires two kinds of flippings (normal and double-flipped), and it roughly corresponds to a regular polygonal tiling – in this case, the square. Take a square of 9 squares, and attach one square on each edge. That is the tiling for this fractal curve. Since Mandelbrot referred to space-filling curves as "Peano Curves", and also referred to the Gosper Curve as "Peano-Gosper", I call this one the "13-Peano-Gosper".

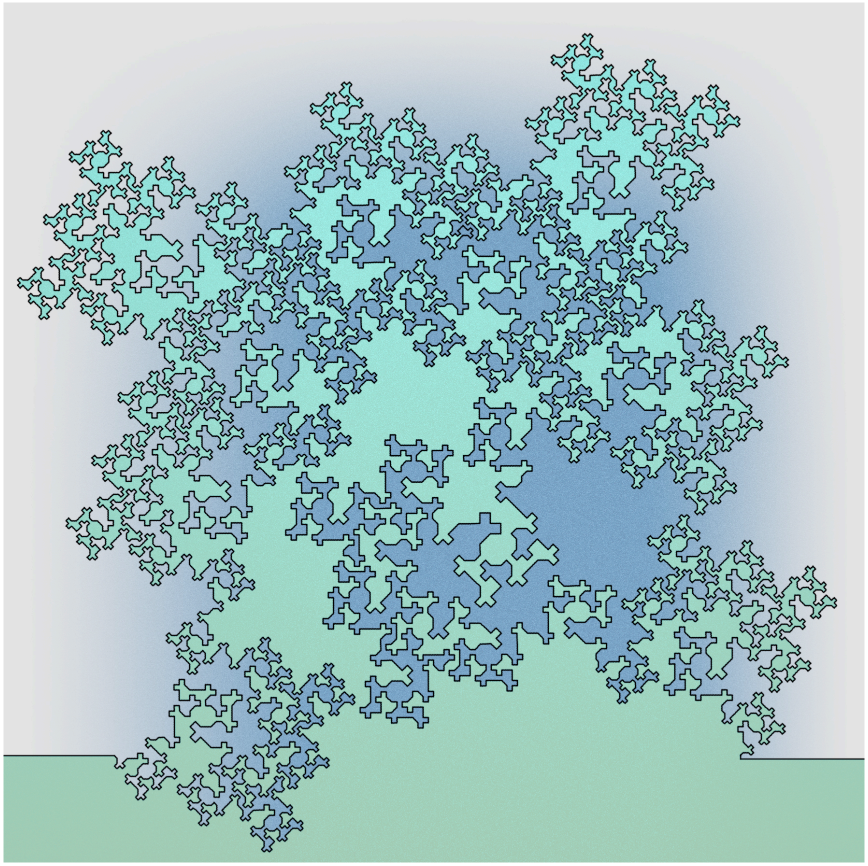




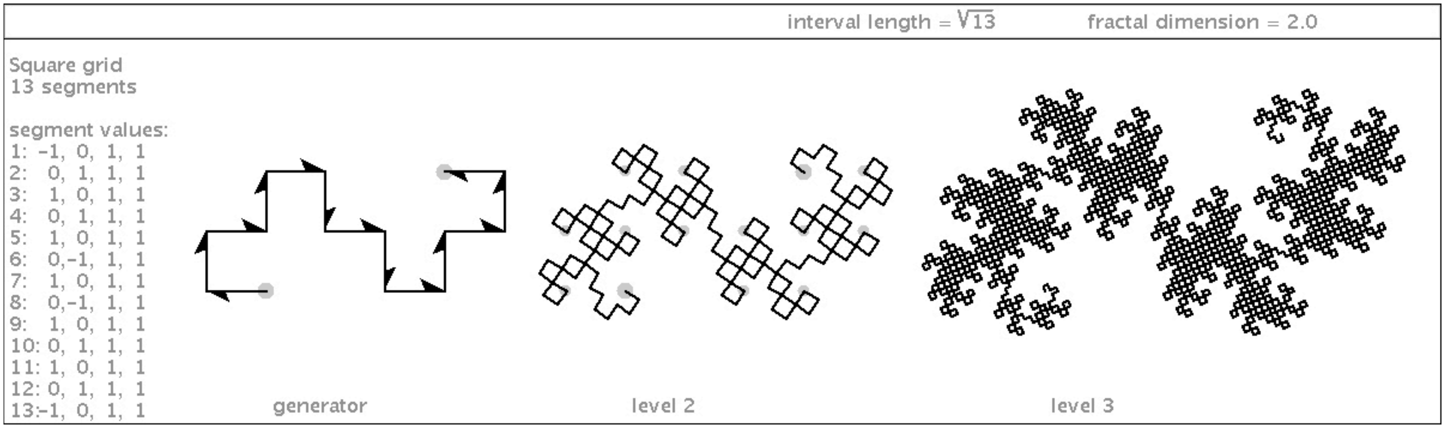
There are several variations of this curve with dimension less than 2. I'll show you two on the next page, followed by a filled-in version of the second one.

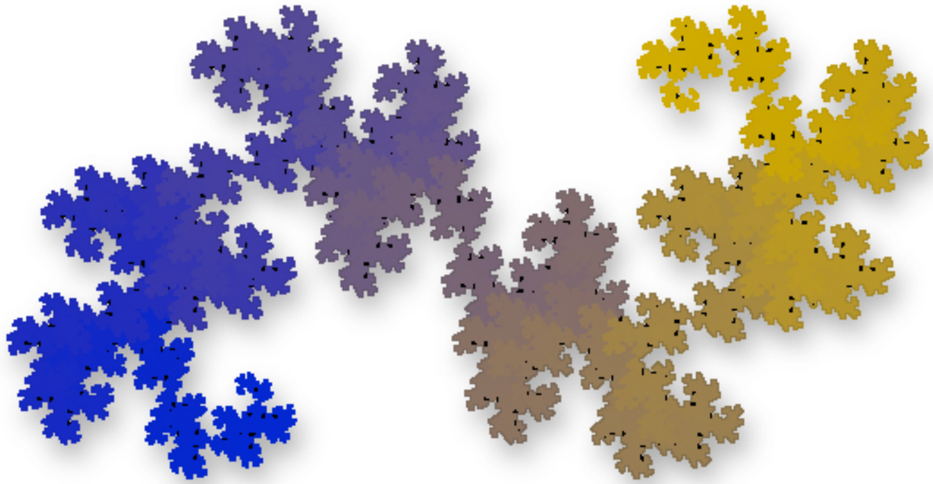

interval length = √13
fractal dimension = 1.8697408
Square grid
10 segments
segment values:
1: 0, 1,-1,-1
2: 0, 1,-1,-1
3: -1, 0, 1, 1
4: 0, 1, 1, 1
5: 1, 1,-1,-1
6: 1, 0, 1, 1
7: 0,-1, 1, 1
8: 0,-1, 1, 1
9: 1, 0,-1,-1
10: 1, 0, 1, 1
generator
level 3


interval length = √13
fractal dimension = 1.8697408
Square grid
10 segments
segment values:
1: 0, 1,-1,-1
2: 0, 1, 1, 1
3: -1, 0,-1,-1
4: 0, 1, 1, 1
5: 1, 0, 1, 1
6: 0, 1,-1,-1
7: 1, 0, 1, 1
8: 0,-1, 1, 1
9: 1,-1,-1,-1
10: 1, 0, 1, 1
level 3 (rounded corners)
generator

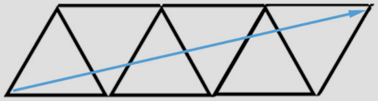


I would like to nominate this as the token Palindrome Dragon of the √13 square grid family:





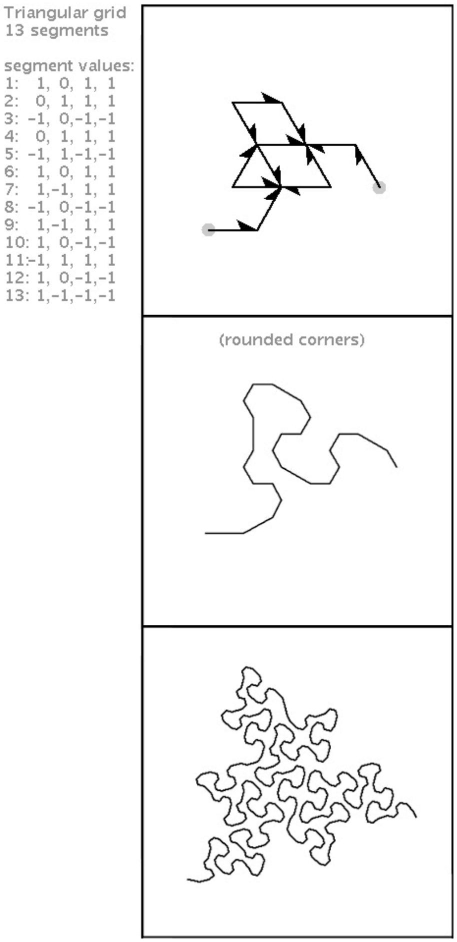
√13

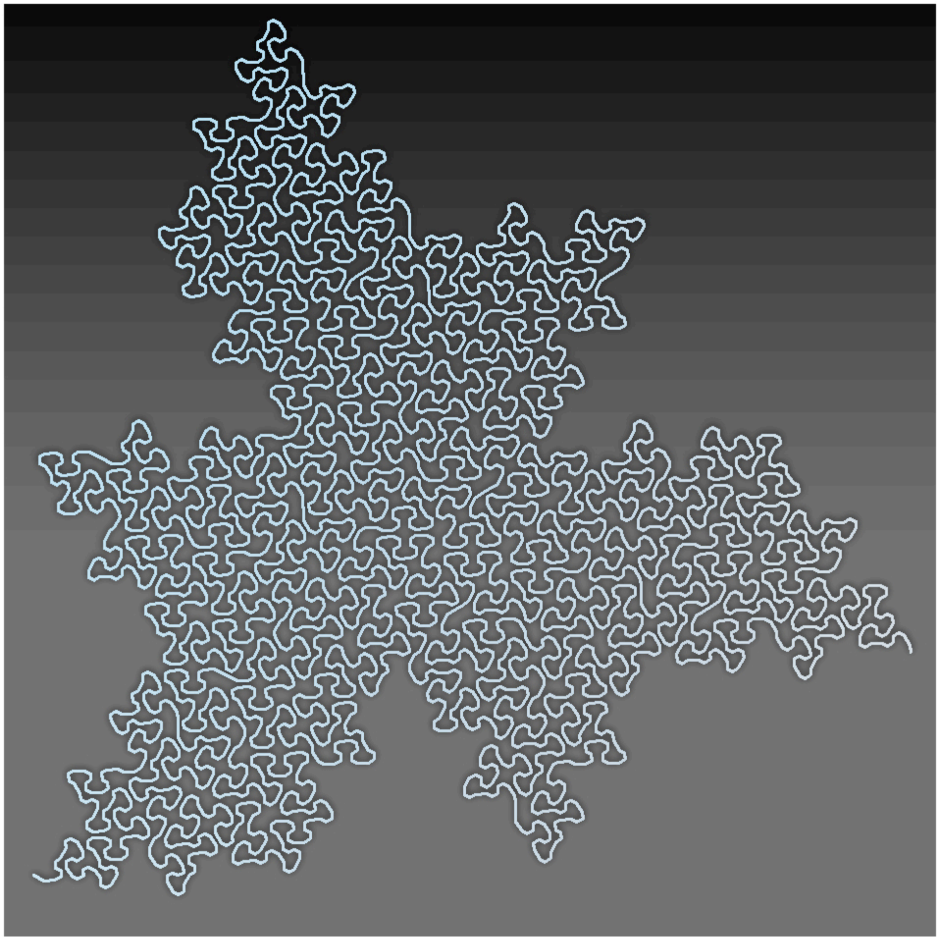


There is a Triangular Grid √13 Family as well as a Square Grid one. Let's check out a few specimens. Here is a curve I discovered which is a partial gridfiller. I'll show it with an expanded version of the diagram scheme:

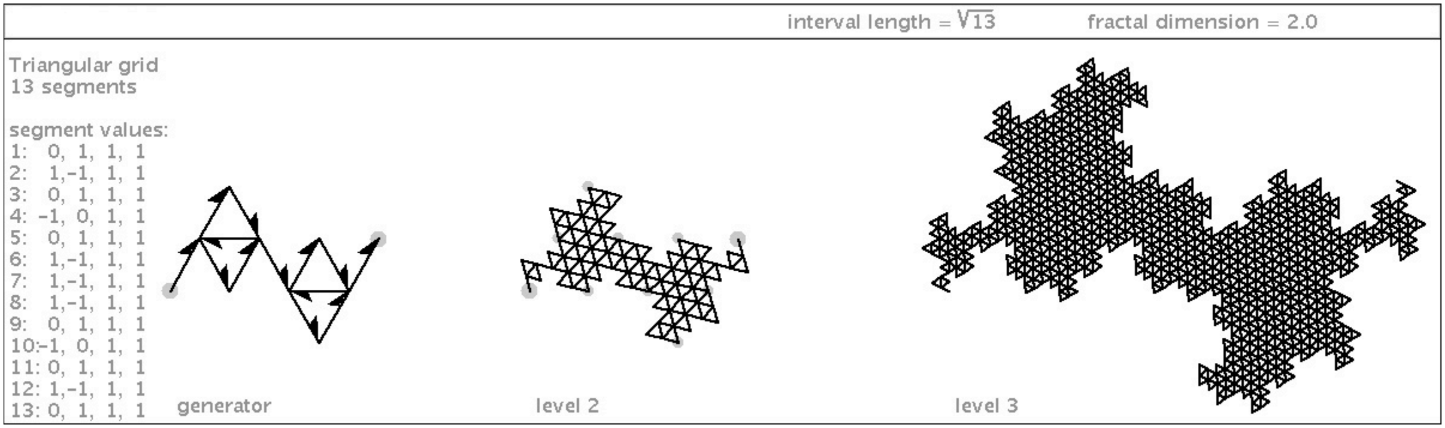
interval length = √13

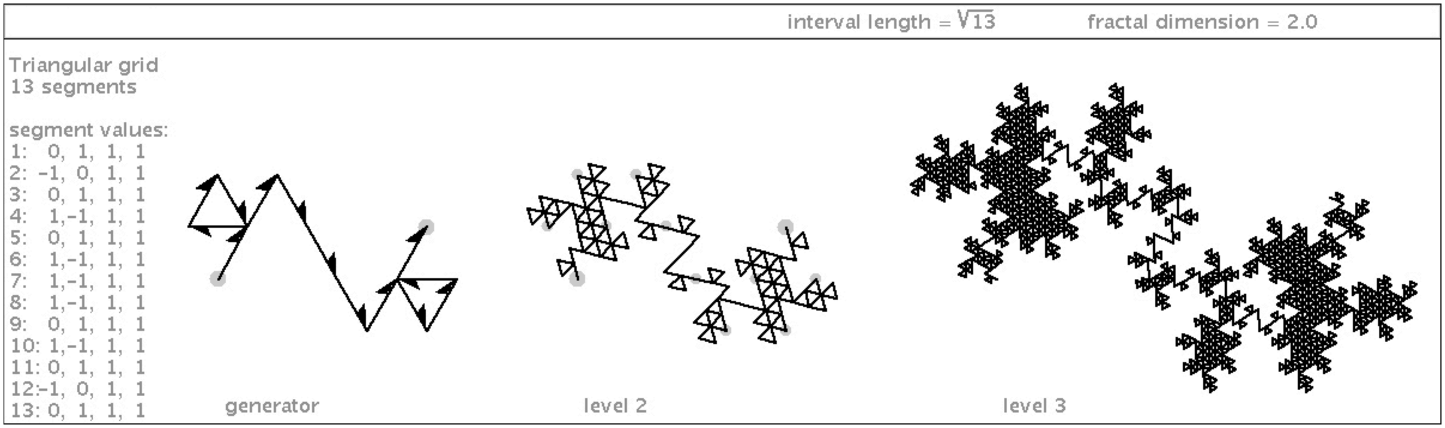
fractal dimension = 2.0



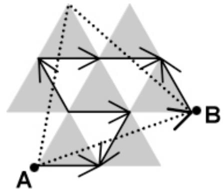


There are a few curious-looking palindrome dragons in this family:



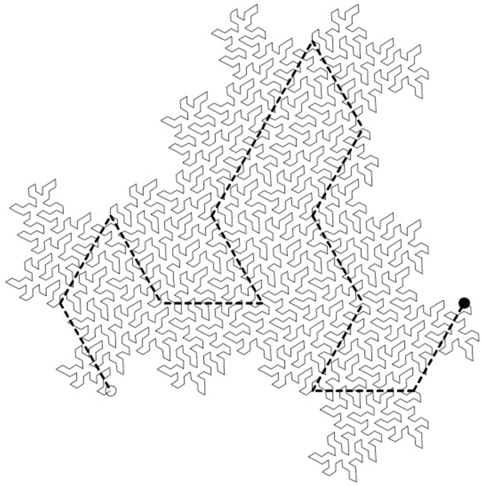


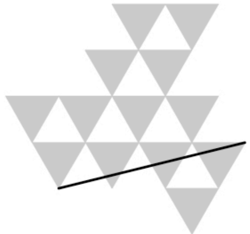
Generalized Gosper Curves Discovered by Fukuda, Shimizu, and Nakamura

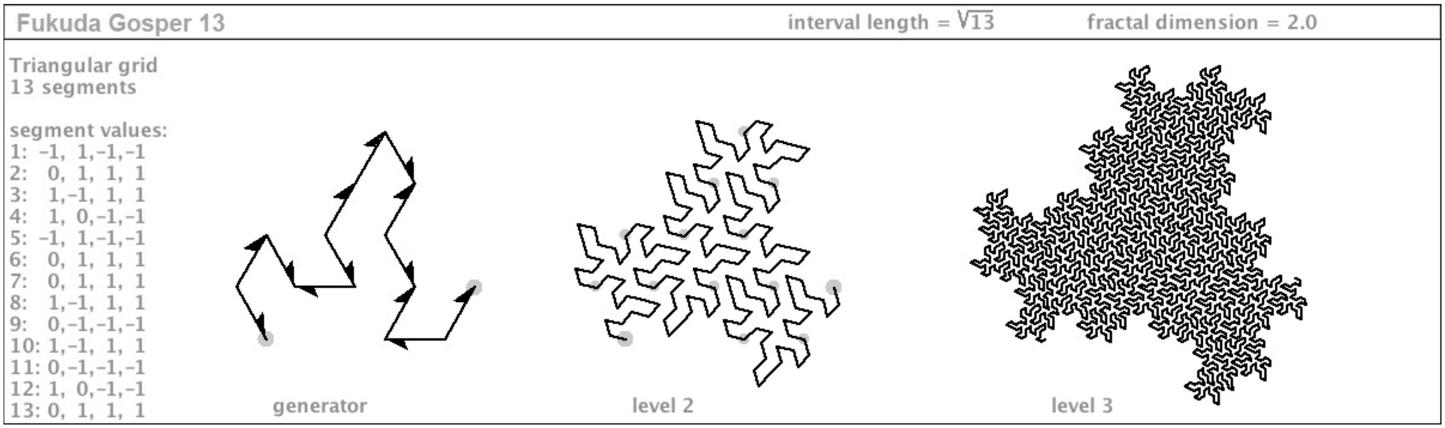


Fractal explorers in Japan devised a search algorithm that finds what they call "generalized Gosper curves" (Fukuda, et al [4]). These are plane-filling curves that use the scheme I described on page 97. Basically, the Gosper Curve is drawn over a grid of seven triangles arranged in a checkerboard fashion. The picture at right shows how the interval length stretches from A to B. Each of the seven triangles maps to the larger triangle outlined with dotted lines.

This ingenious scheme permits the search algorithm to find an infinite series of generalized Gosper Curves. The diagram at right shows how a similar assembly of triangles can be used to identify a generalized Gosper curve that is a member of the √13 triangle grid family. Below I show how this curve fits within my scheme. Later on I'll show you another one of the many curves that these explorers have uncovered.

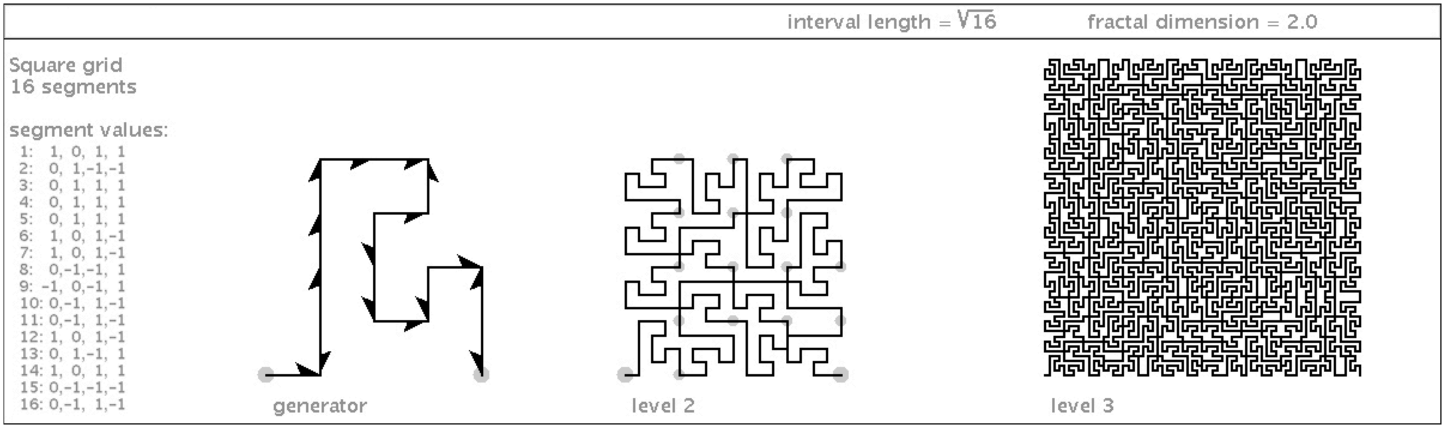


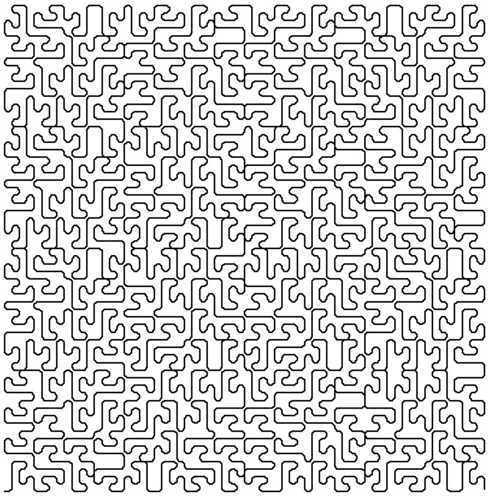


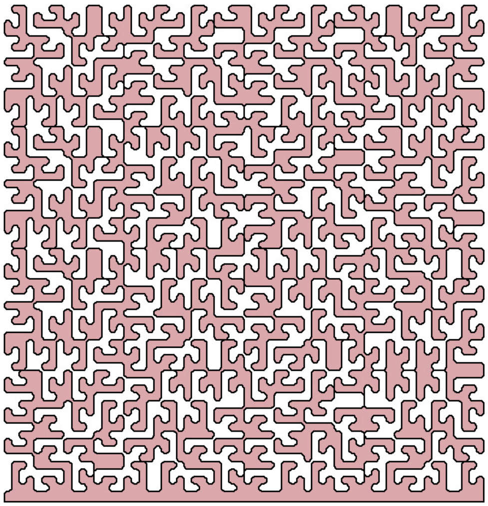


√16

One way to express a square number S is to draw a square made up of S smaller squares. I've shown you fractal curves of the √4 and √9 families where square regions are filled. Now we come to the next square number: 16. Below is a curve of dimension 2 that is a partial gridfiller, and it is also partially edge self-touching. On the next page are two specimens with dimension < 2 that are “trying” to fill a square, and a 2D specimen with 4-fold symmetry.



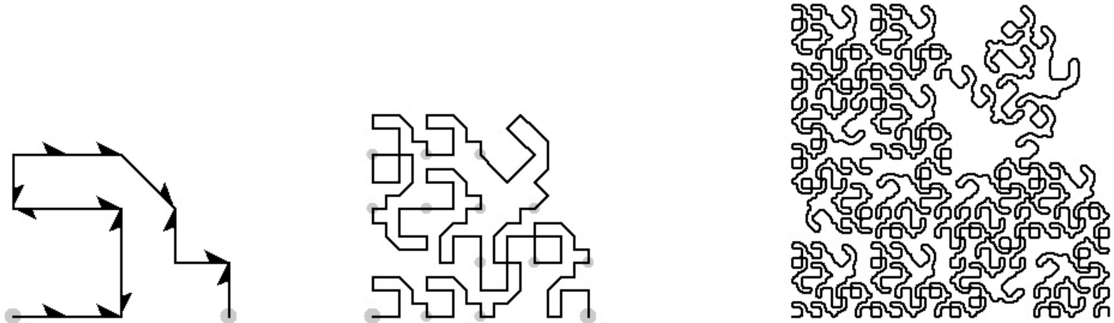




interval length = $\sqrt{16}$ fractal dimension = 1.9036775

Square grid
13 segments

segment values:
1: 1, 0, 1, 1
2: 1, 0, 1, 1
3: 0, 1,-1,-1
4: 0, 1, 1, 1
5: -1, 0,-1,-1
6: -1, 0, 1, 1
7: 0, 1,-1,-1
8: 1, 0, 1, 1
9: 1, 0, 1, 1
10: 1,-1, 1, 1
11: 0,-1,-1,-1
12: 1, 0, 1, 1
13: 0,-1,-1,-1

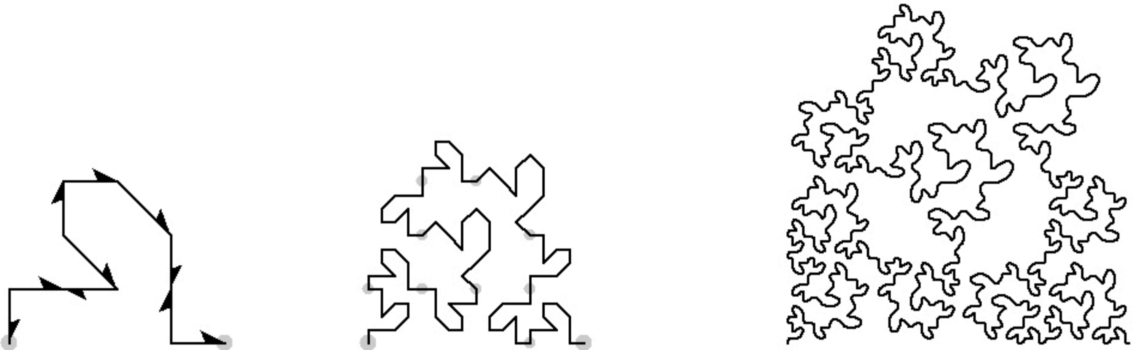


generator level 2 level 3 (rounded corners)

interval length = $\sqrt{16}$ fractal dimension = 1.7924813

Square grid
10 segments

segment values:
1: 0, 1,-1,-1
2: 1, 0, 1, 1
3: 1, 0,-1,-1
4: -1, 1,-1,-1
5: 0, 1, 1, 1
6: 1, 0, 1, 1
7: 1,-1, 1, 1
8: 0,-1, 1, 1
9: 0,-1,-1,-1
10: 1, 0, 1, 1

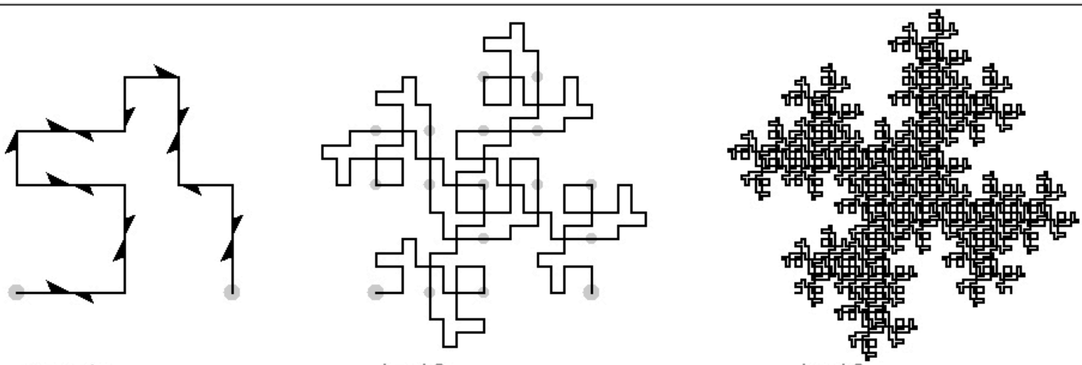


generator level 2 level 3 (rounded corners)

interval length = $\sqrt{16}$ fractal dimension = 2.0

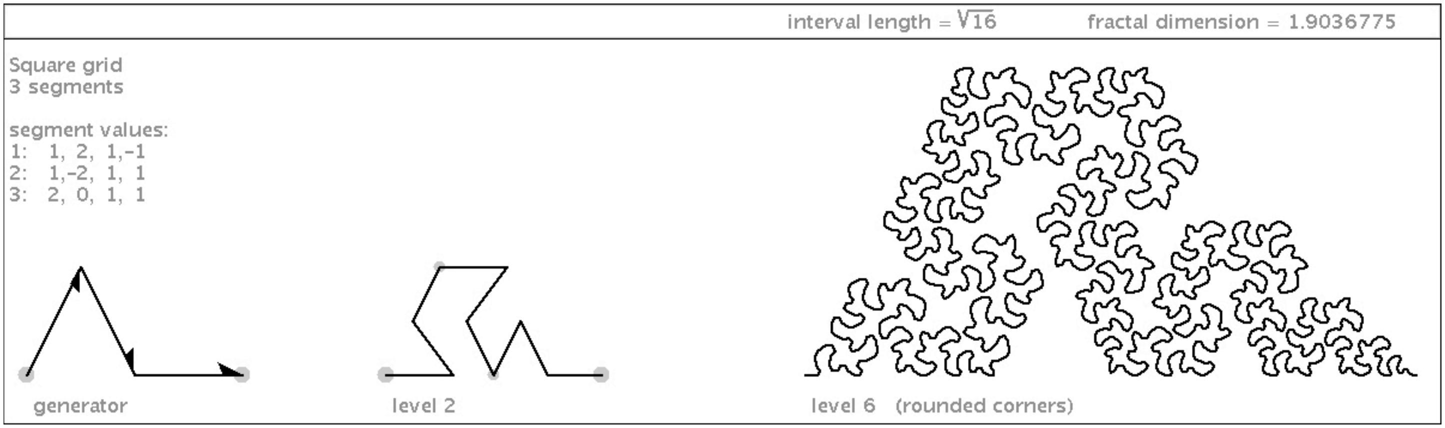
Square grid
16 segments

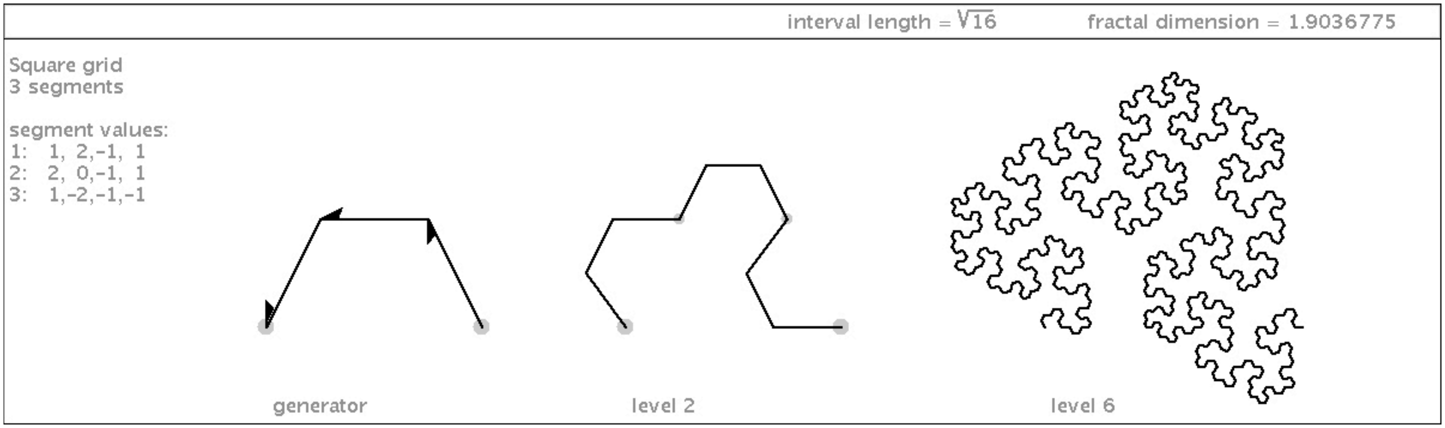
segment values:
1: 1, 0, 1, 1
2: 1, 0,-1,-1
3: 0, 1, 1, 1
4: 0, 1,-1,-1
5: -1, 0, 1, 1
6: -1, 0,-1,-1
7: 0, 1, 1, 1
8: 1, 0, 1, 1
9: 1, 0,-1,-1
10: 0, 1,-1,-1
11: 1, 0, 1, 1
12: 0,-1, 1, 1
13: 0,-1,-1,-1
14: 1, 0,-1,-1
15: 0,-1, 1, 1
16: 0,-1,-1,-1



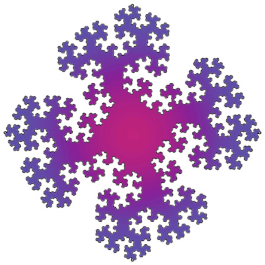
generator level 2 level 3

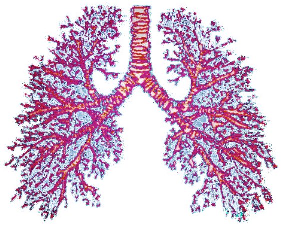
The following two generators of the √16 square grid family have only three segments. They do not correspond to hexagonal or triangular grid lines, even though they may appear to upon first glance – they are indeed inhabitants of the square grid. In both curves, the upward-sloping segments traverse across the diagonal of two squares, thus having lengths of √5. The horizontal segments have lengths of 2.





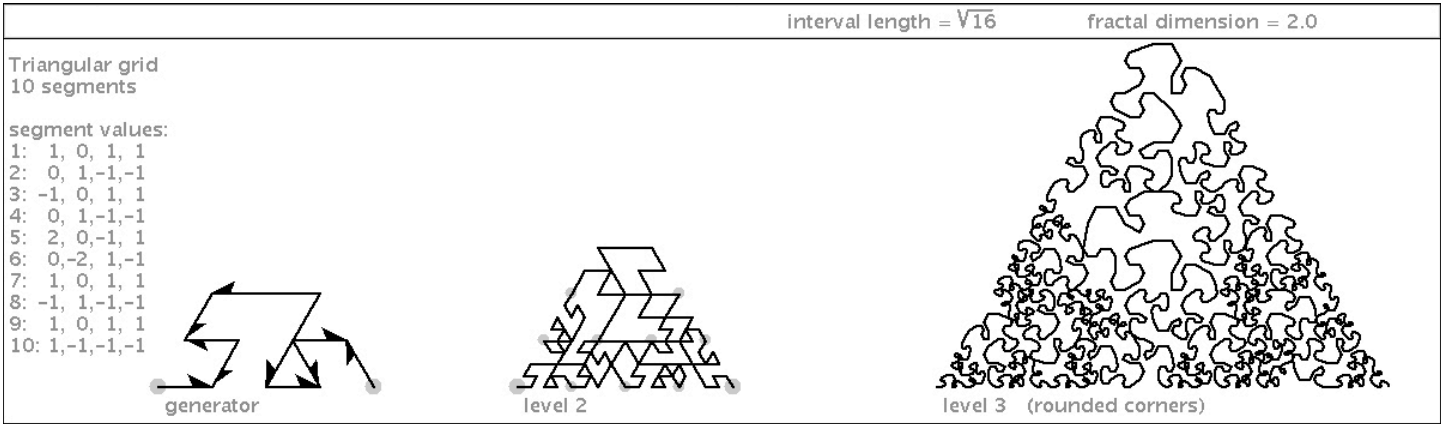
This curve can be copied four times and placed on a square initiator to create the closed curve shown at right. There is a vague resemblance to the structure of bronchial tubes.

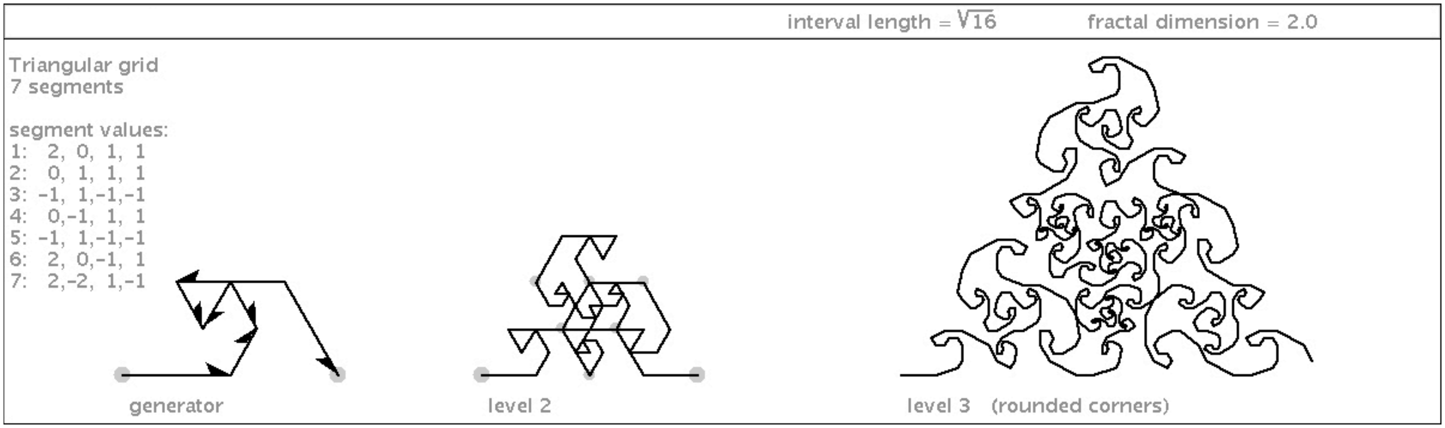


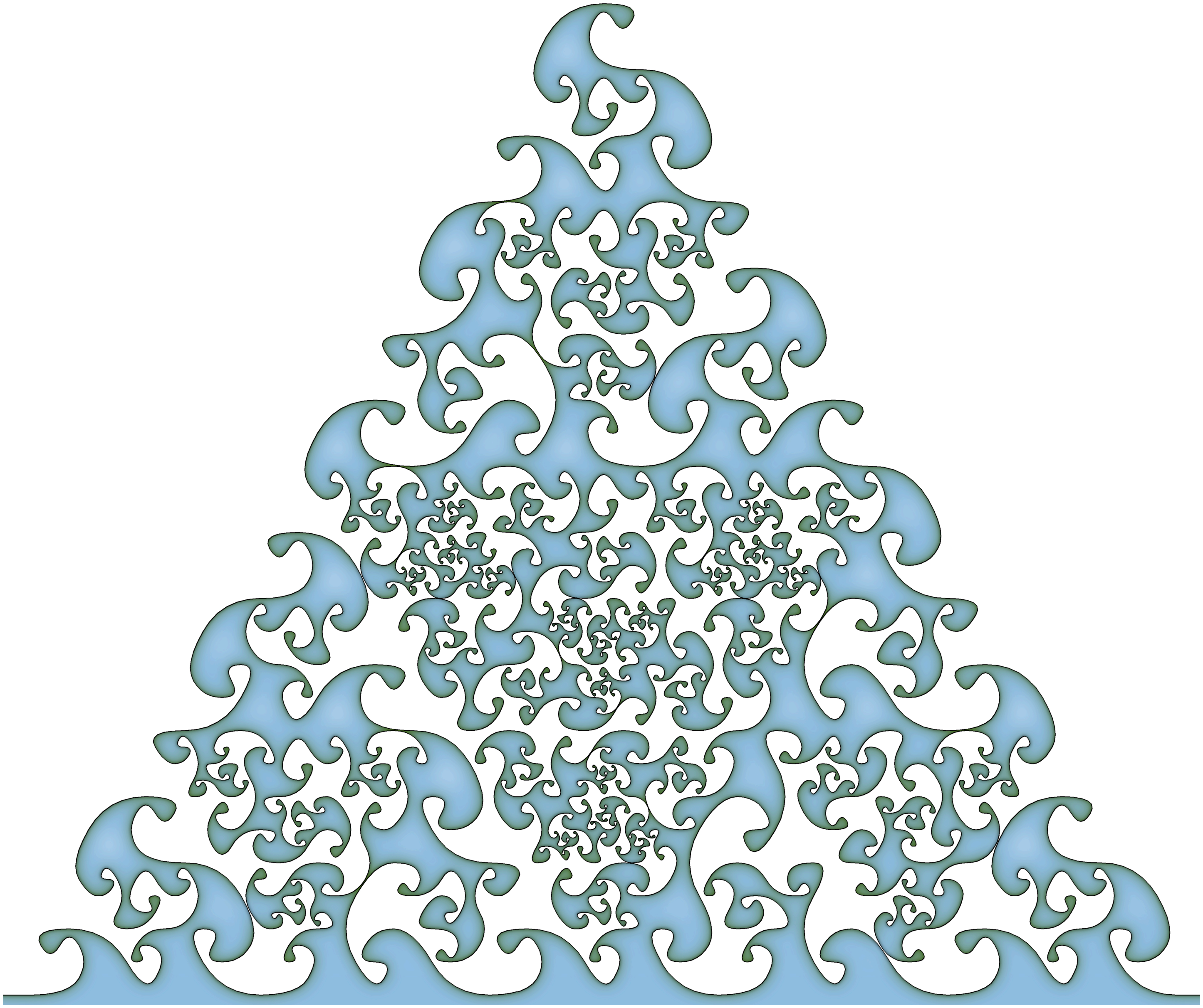


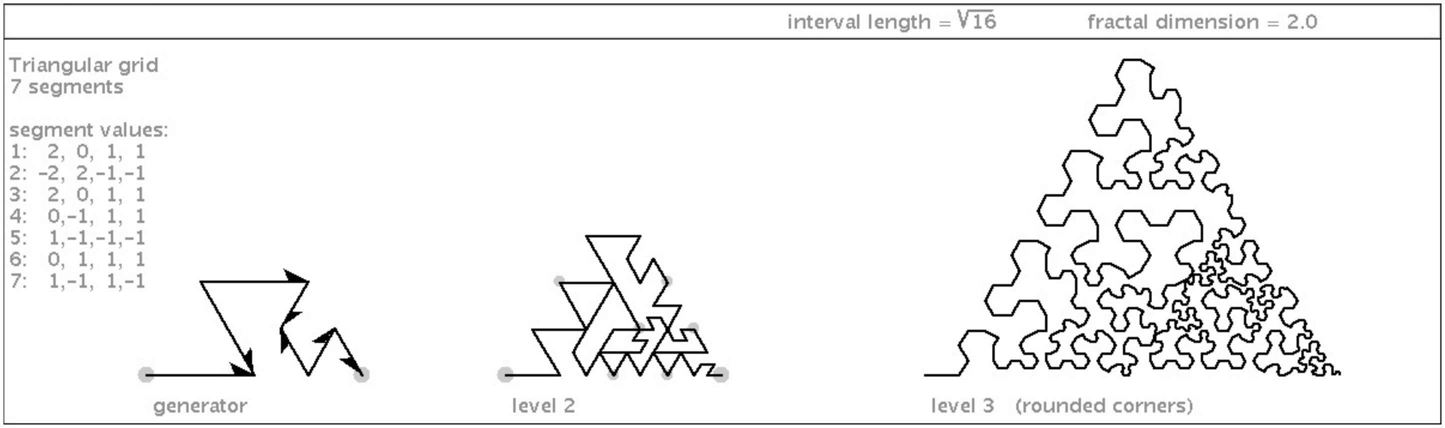
$\sqrt{16}$

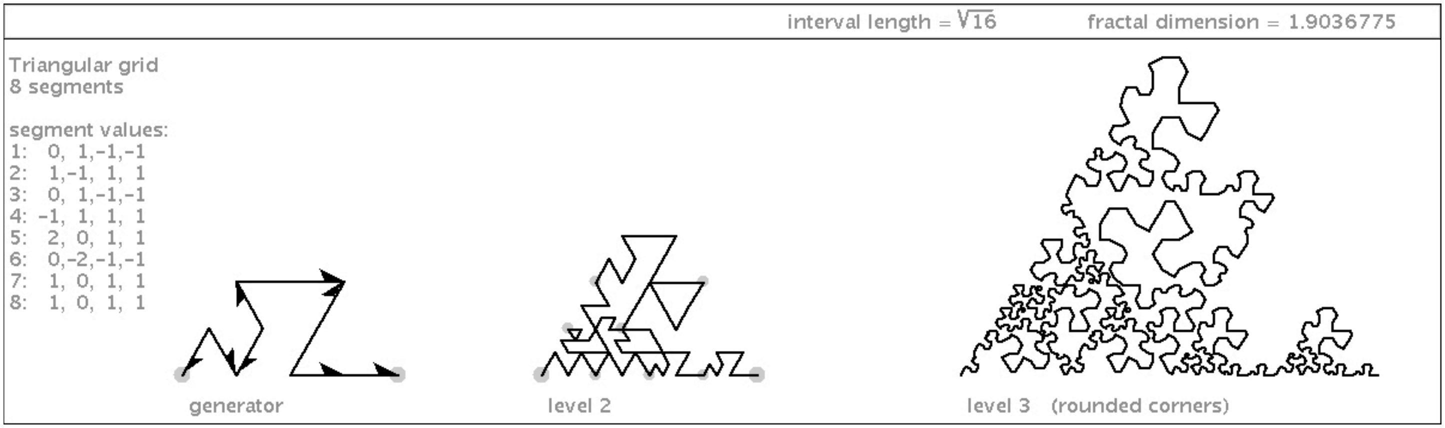
Like the √16 square grid family, the √16 triangle grid family has many spacefilling curves that pertile equilateral polygons – in this case, triangles. I would like to show you a few. None of these examples are self-avoiders…but when rendered with rounded corners, they reveal some unique characteristics. The next four specimens each have a segment of length 2 that lies horizontally across the middle of the 4x4 triangle that houses the generator. This horizontal segment is responsible for extending the curve upward towards the pinnacle – increasingly with each fractal level.



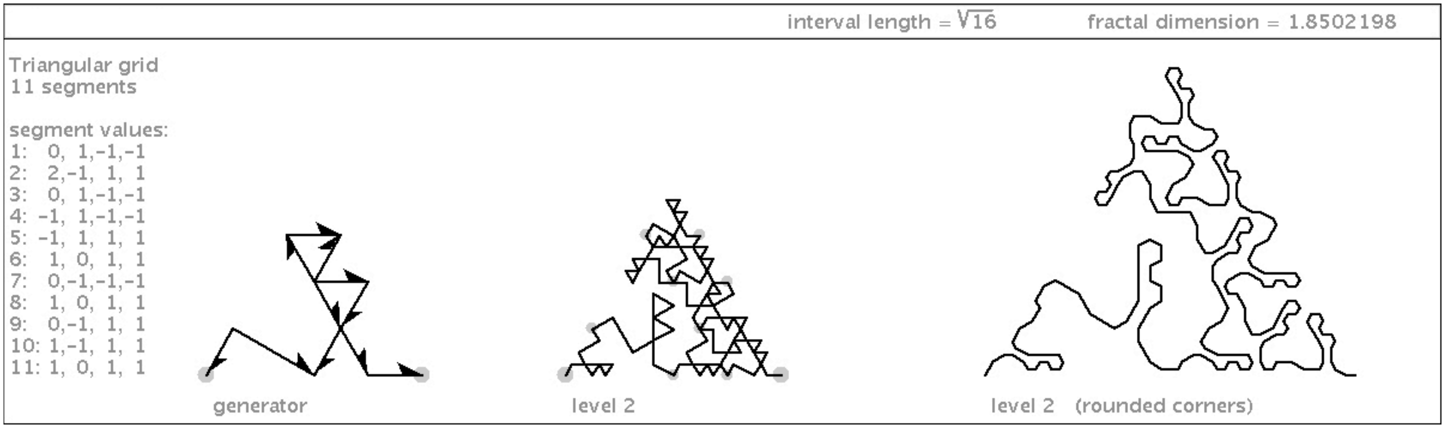


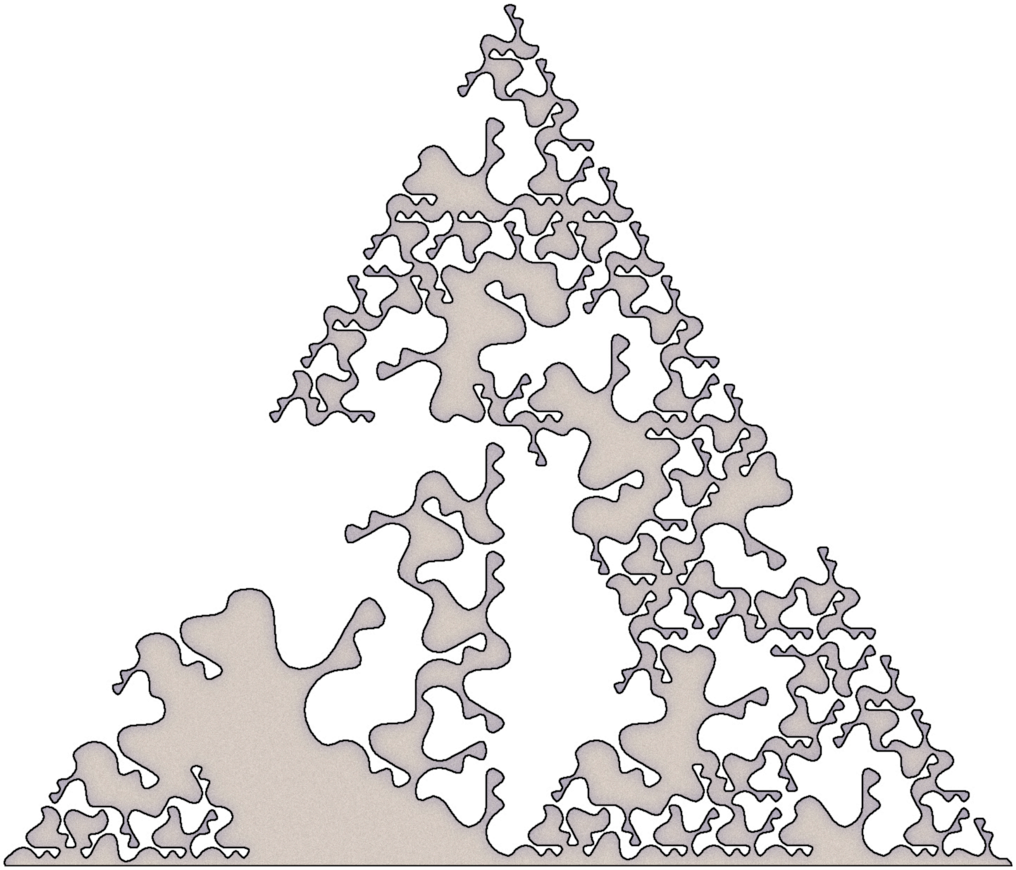






On the next three pages are triangular specimens that I have rendered in color. The last one has the top one-fourth of the triangle rotated 180 degrees and enlarged, to its right.

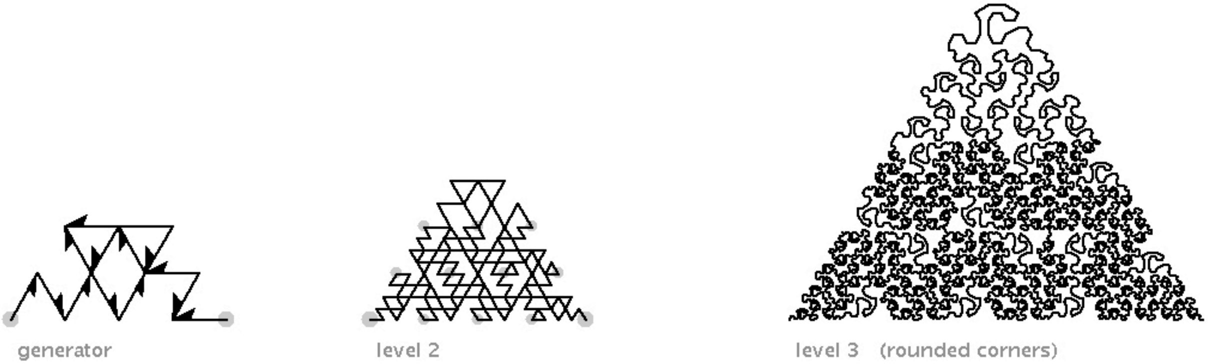

interval length = $\sqrt{16}$
fractal dimension = 1.8502198
Triangular grid
11 segments
segment values:
1: 0, 1,-1,-1
2: 2,-1, 1, 1
3: 0, 1,-1,-1
4: -1, 1,-1,-1
5: -1, 1, 1, 1
6: 1, 0, 1, 1
7: 0,-1,-1,-1
8: 1, 0, 1, 1
9: 0,-1, 1, 1
10: 1,-1, 1, 1
11: 1, 0, 1, 1
generator
level 2
level 2 (rounded corners)

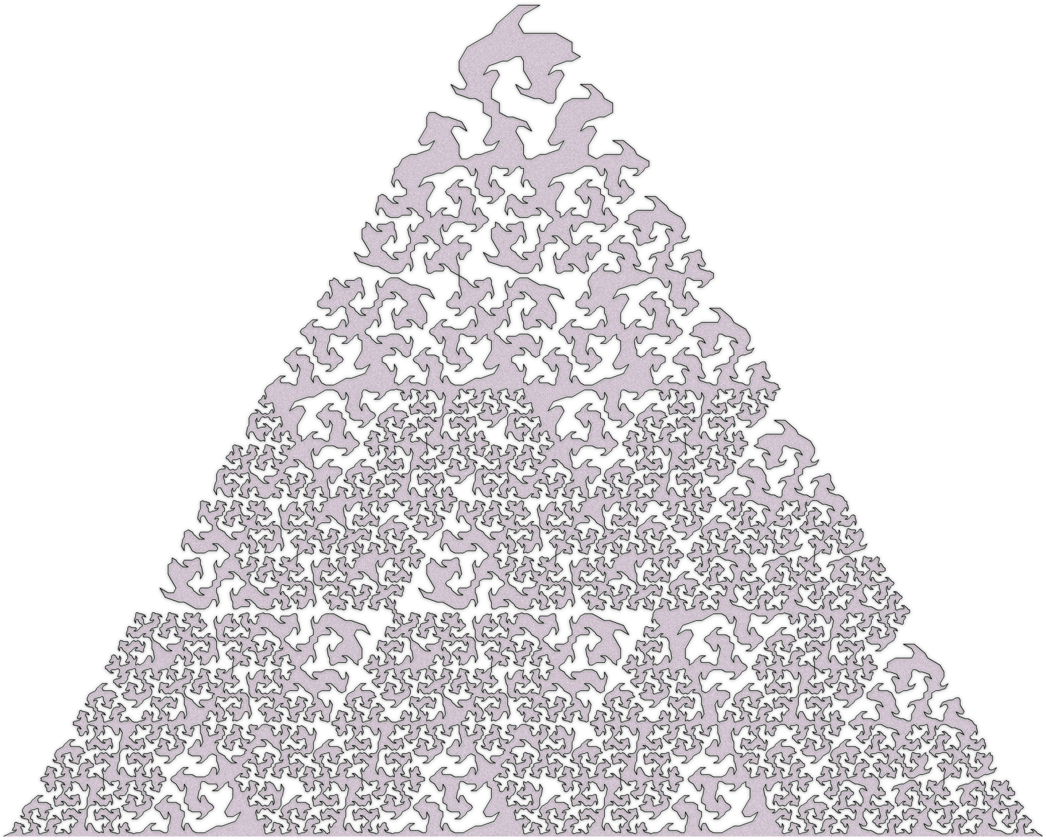


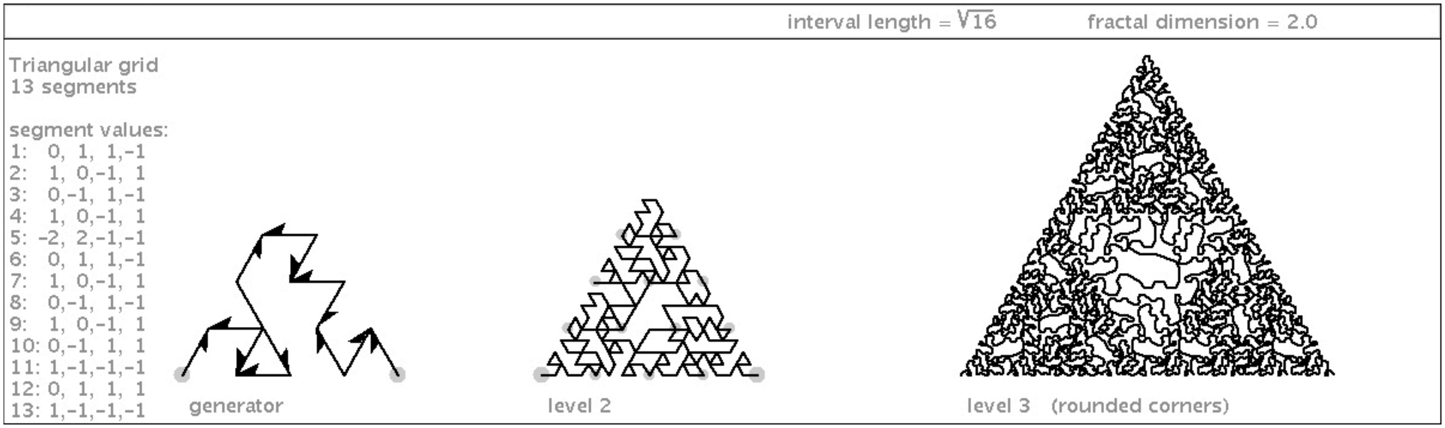
interval length = $\sqrt{16}$ fractal dimension = 2.0

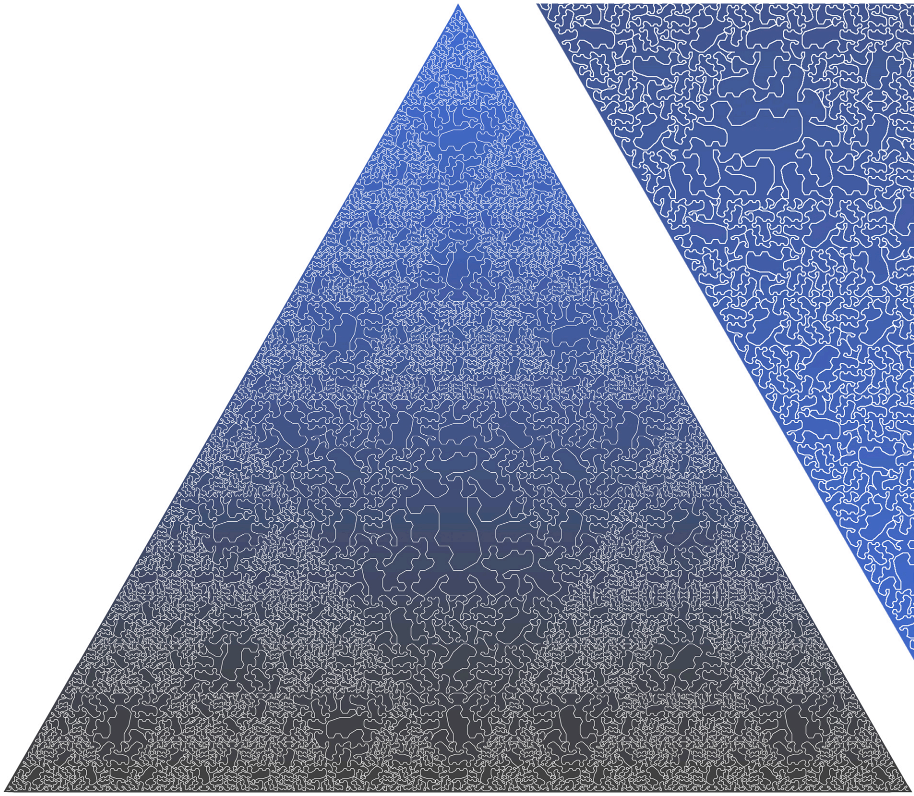
Triangular grid
13 segments

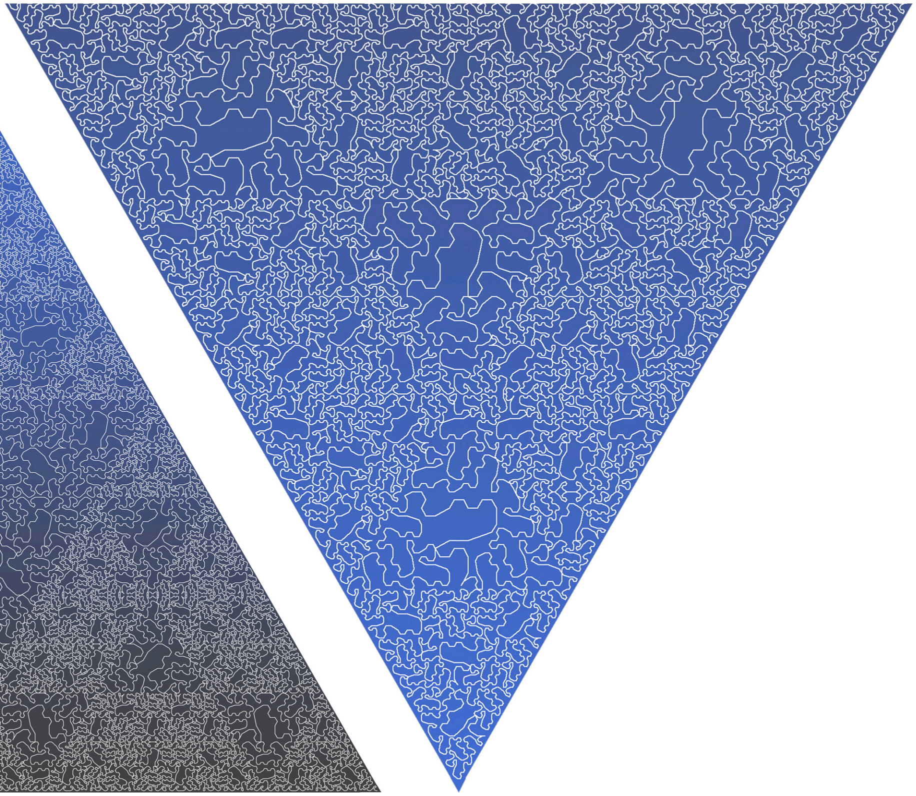
segment values:
1: 0, 1, 1,-1
2: 1,-1, 1, 1
3: 0, 1, 1,-1
4: -1, 1, 1, 1
5: 2, 0,-1, 1
6: 0,-1, 1,-1
7: -1, 1, 1, 1
8: 0,-1, 1,-1
9: 1,-1, 1, 1
10: 0, 1, 1,-1
11: 1, 0,-1, 1
12: 0,-1, 1,-1
13: 1, 0,-1, 1



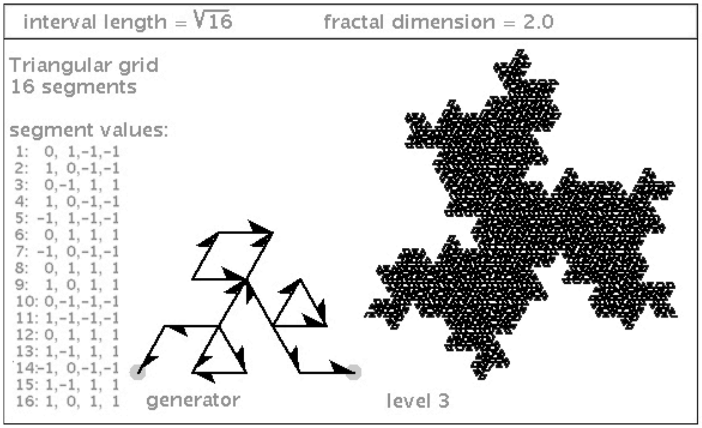


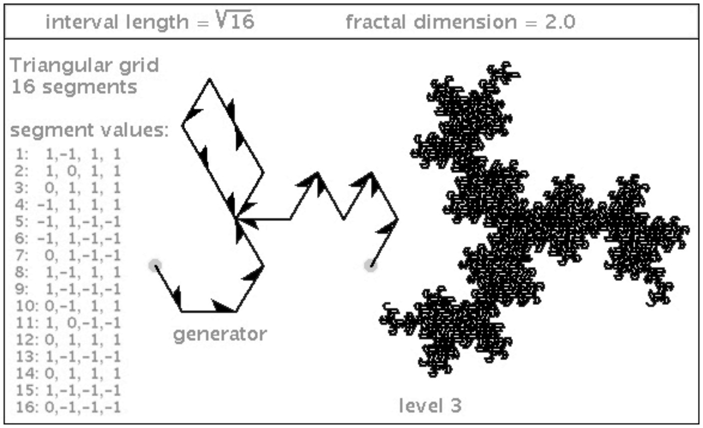

interval length = $\sqrt{16}$
fractal dimension = 2.0
Triangular grid
13 segments
segment values:
1: 0, 1, 1,-1
2: 1, 0,-1, 1
3: 0,-1, 1,-1
4: 1, 0,-1, 1
5: -2, 2,-1,-1
6: 0, 1, 1,-1
7: 1, 0,-1, 1
8: 0,-1, 1,-1
9: 1, 0,-1, 1
10: 0,-1, 1, 1
11: 1,-1,-1,-1
12: 0, 1, 1, 1
13: 1,-1,-1,-1
generator
level 2
level 3 (rounded corners)

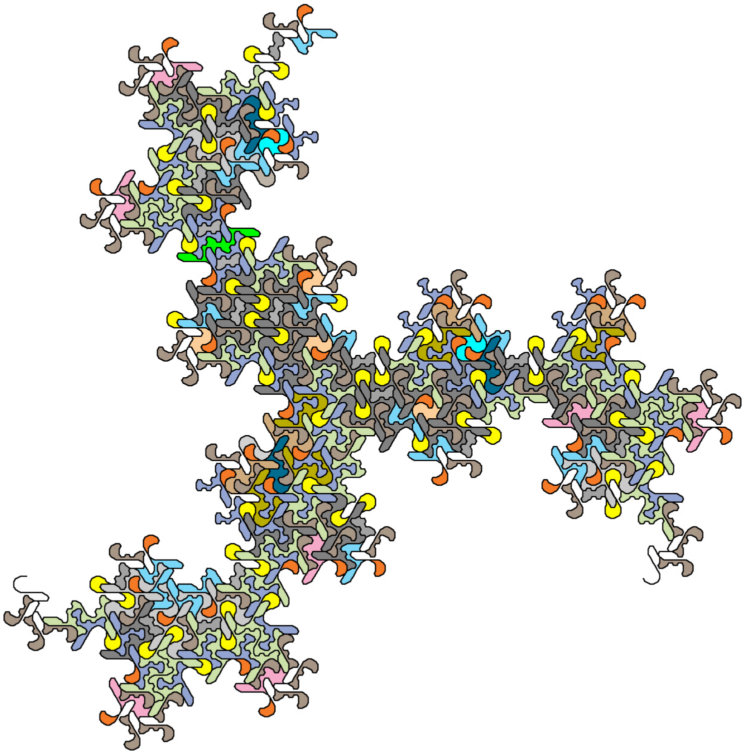




Here are some non-triangle specimens with three-way symmetry. These tightly-coiled curves are of the dense variety that evade being pulled apart to breathe by way of rounded corners. But they do have interesting chambers. The second specimen is rendered with its various chambers colored in a playful manner.







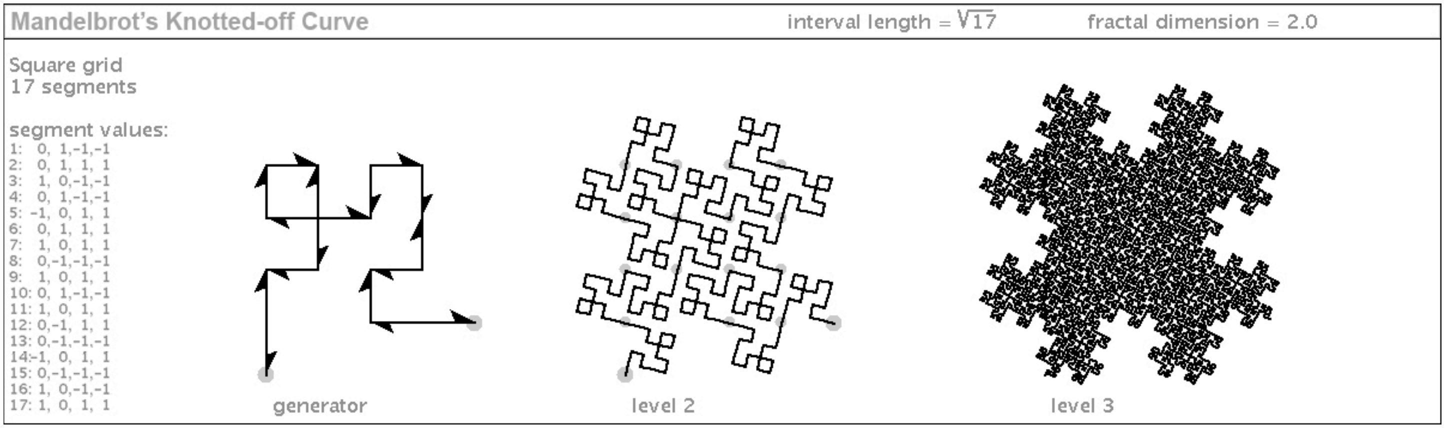
$\sqrt{17}$ ... and beyond

The number of plane-filling specimens in the √2 family is 2 (I am not counting Cesàro). The number of plane-filling specimens in the √3 family (by my estimate) is 10. As we climb the family tree, the number of specimens increases non-linearly, making exhaustive search quite an expensive proposition when many segments are involved. Consider the sheer number of possible generators if each segment has four possible flippings, and connects any two grid points, where its length is less than the interval length. I calculated the number of possible generators having up to 5 segments. Ready?

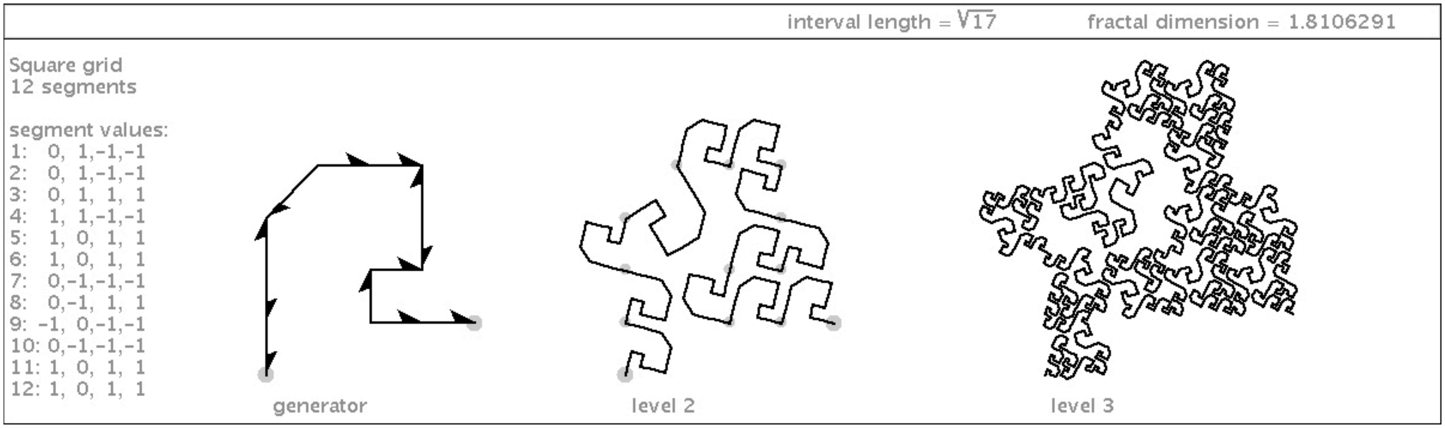
2 seg: **16** 3 seg: **13,824** 4 seg (triangle): **331,776** 4 seg (square): **1,048,576** 5 seg: **254,803,968**

I have done some exploration of this huge space of possibilities, using both hand-drawn diagrams, and computer search algorithms, and have discovered some interesting specimens. Let’s look at some of those now, as well as a few gems previously introduced by Mandelbrot and others.

First, let’s start with a plane-filling curve that Mandelbrot introduced in his book. It is a member of the √17 square grid family, and it is a partial gridfiller. Mandelbrot pointed out that self-avoiding curves are not only more aesthetic; they also make better models of forms found in nature (such as rivers and watershed trees). When a curve is vertex self-touching, it “knots off”. In this example below, the generator itself has a pinched point. This percolates throughout the teragons, in a similar way that I showed you with the self-crossing 7-dragons. Here is Mandelbrot’s Knotted-off √17 curve:

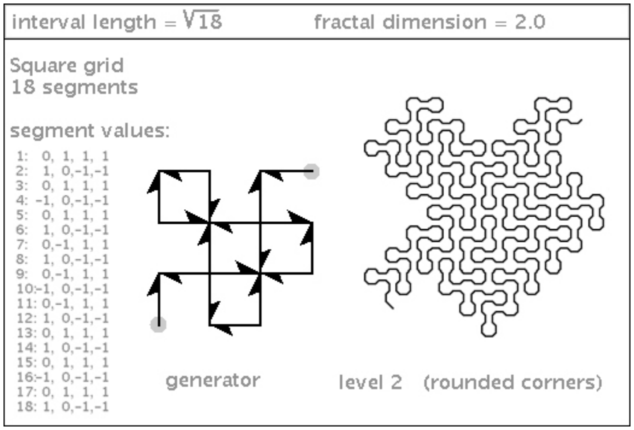


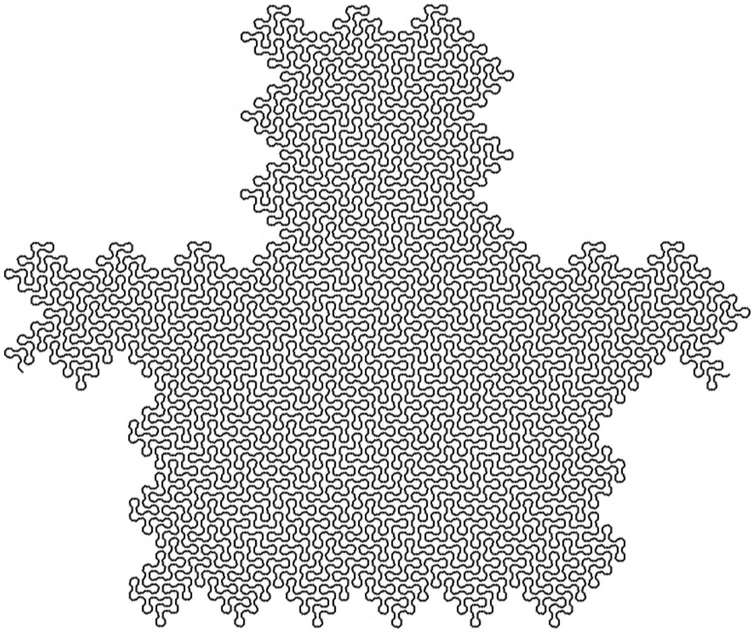
Here is another √17 curve. It is a self-avoider, and it has a dimension of ~1.81.



The √18 Square Grid Family

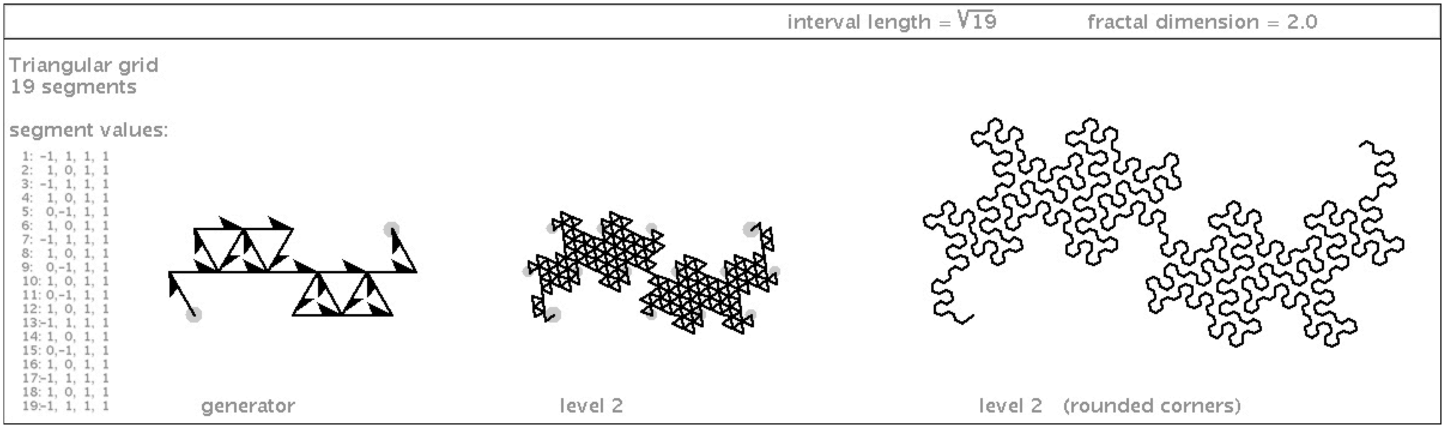
I have one specimen to show of this family. It is a gridfiller. It is shown at right at level 3, rotated by 45 degrees. Personally, I find nothing attractive about this specimen. It looks like a poorly-made gingerbread man. But I am including it in the book…even boring specimens deserve to be seen.

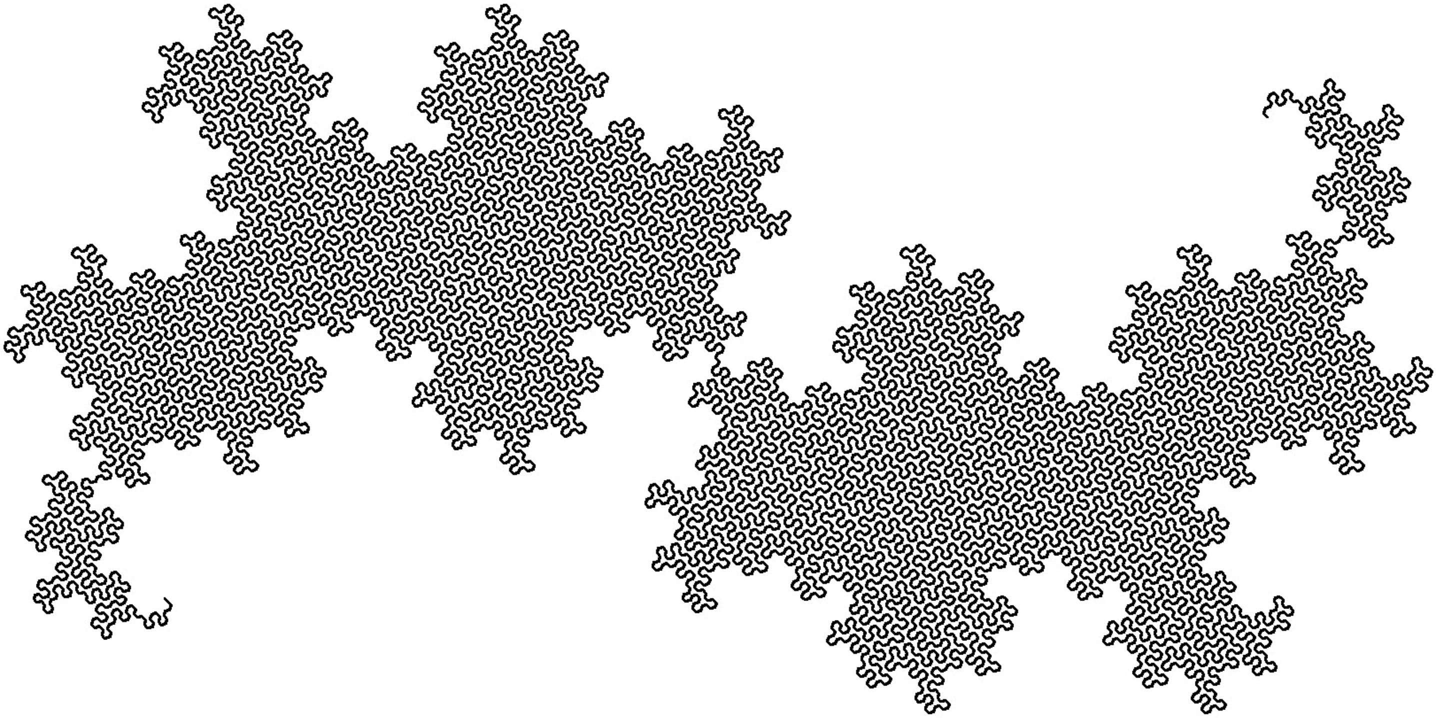


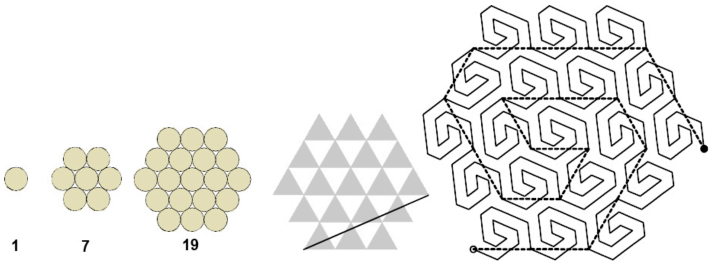


The √19 Triangle Grid Family

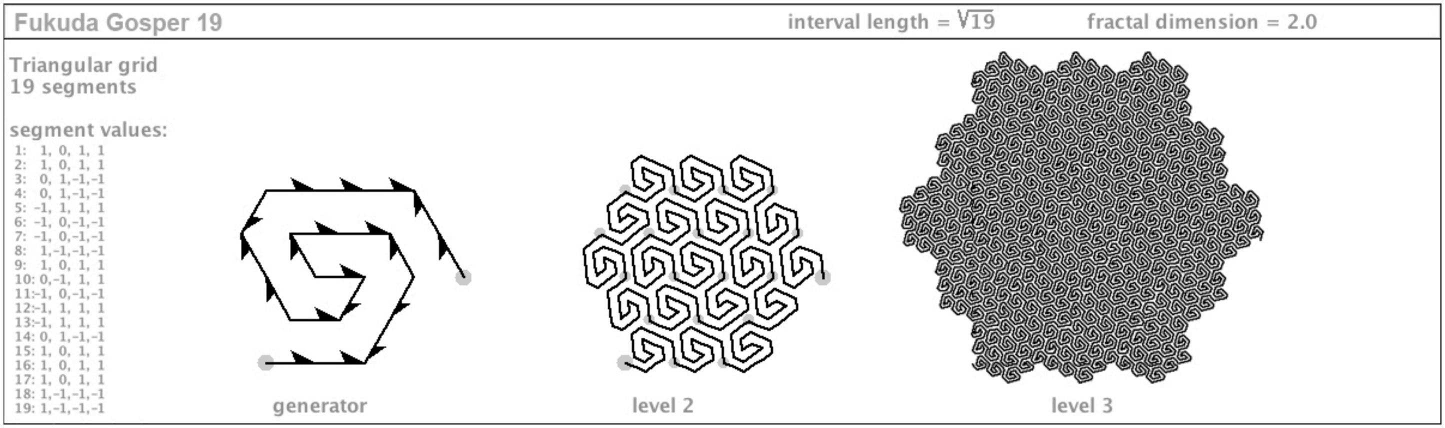
Let's look at a few of the fine specimens of the √19 triangle grid family. I like to start with dragons, and so here is a palindrome dragon with a pinched waist and pinched extremities. It is shown below at level 3 with rounded corners.



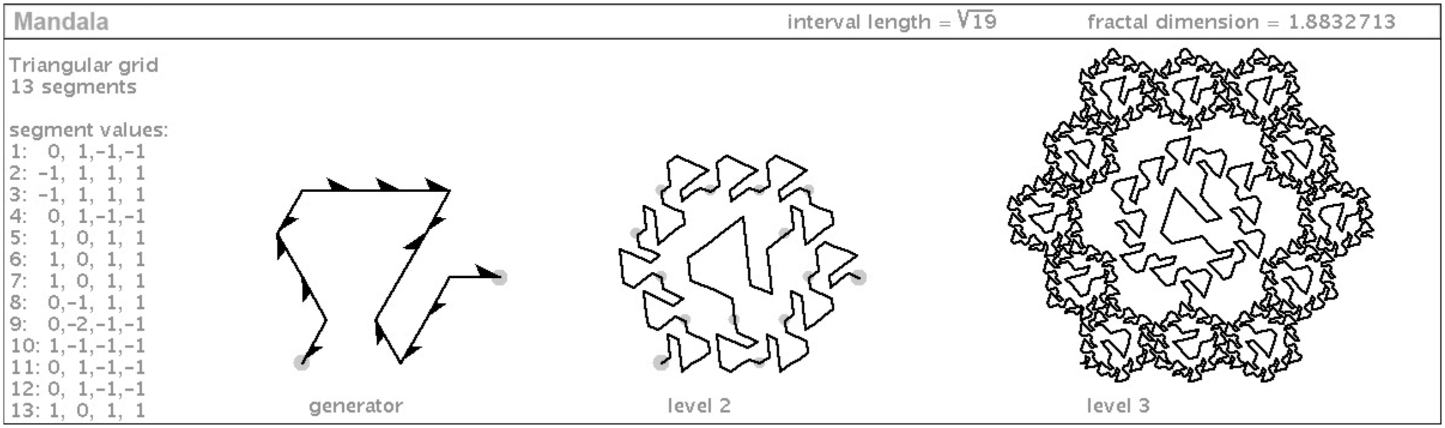


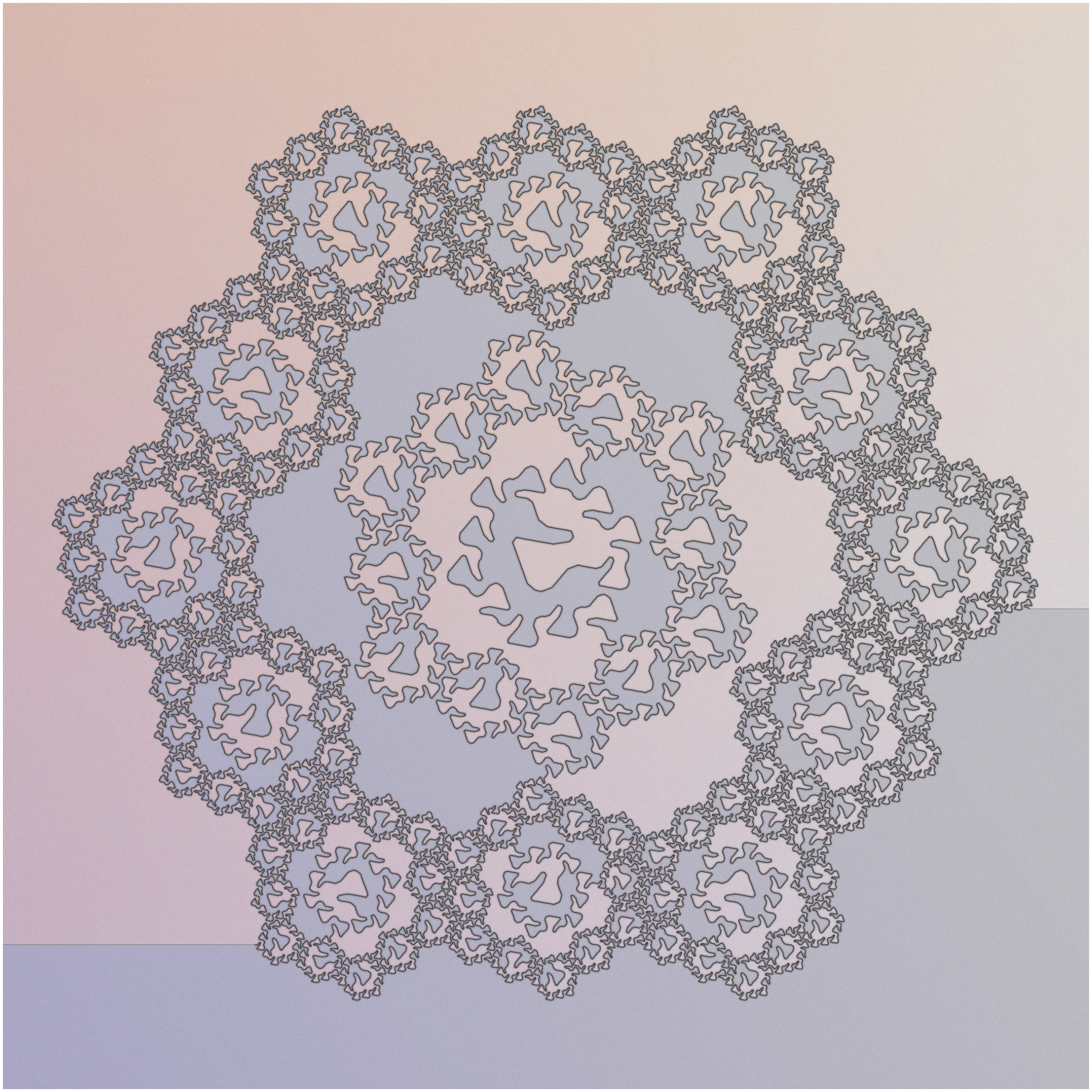


Close-packing of seven circles results in a roughly-hexagonal shape: the basis for the Gosper Curve. If you completely surround these seven circles with a new layer of circles, the total number of circles is increased to 19, as shown at left. This 19-cell hexagonal grid is the basis for another one of the wonderful generalized Gosper curves discovered by Fukuda, et. al [4]. It is shown below.

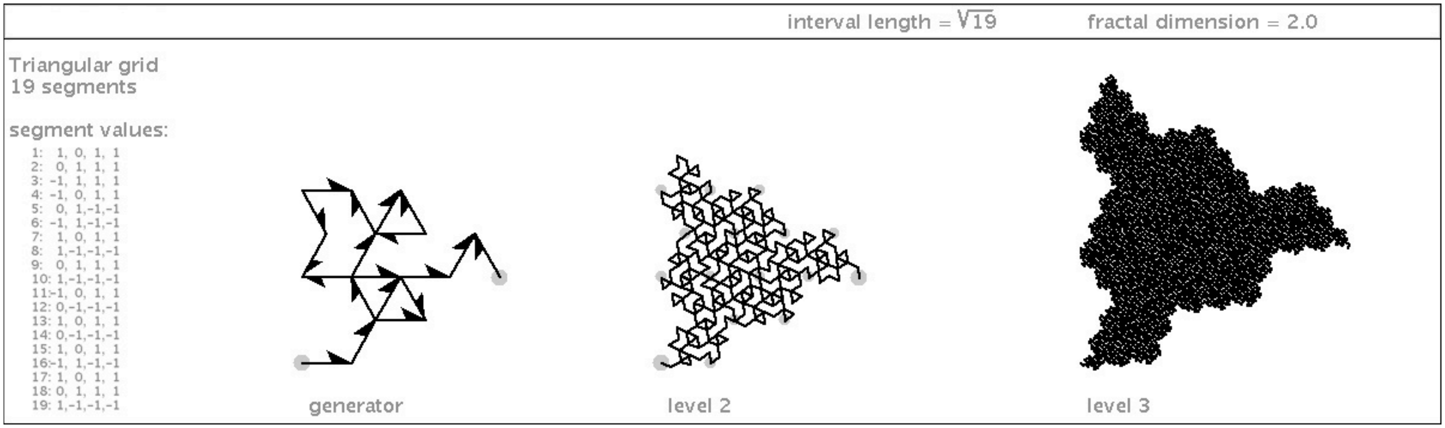


I discovered a curve based on this 19-cell hexagonal theme. Its dimension is less than 2, and so it has lots of open spaces, which gives it some artistic breathing room. I call it "Mandala". It is shown in color on the next page.



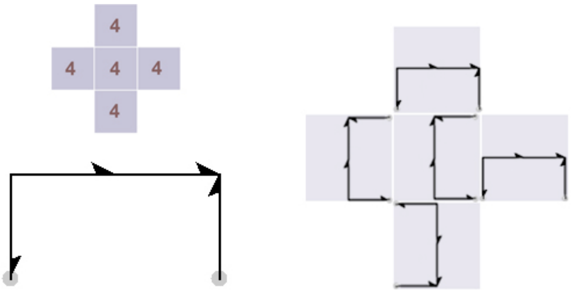


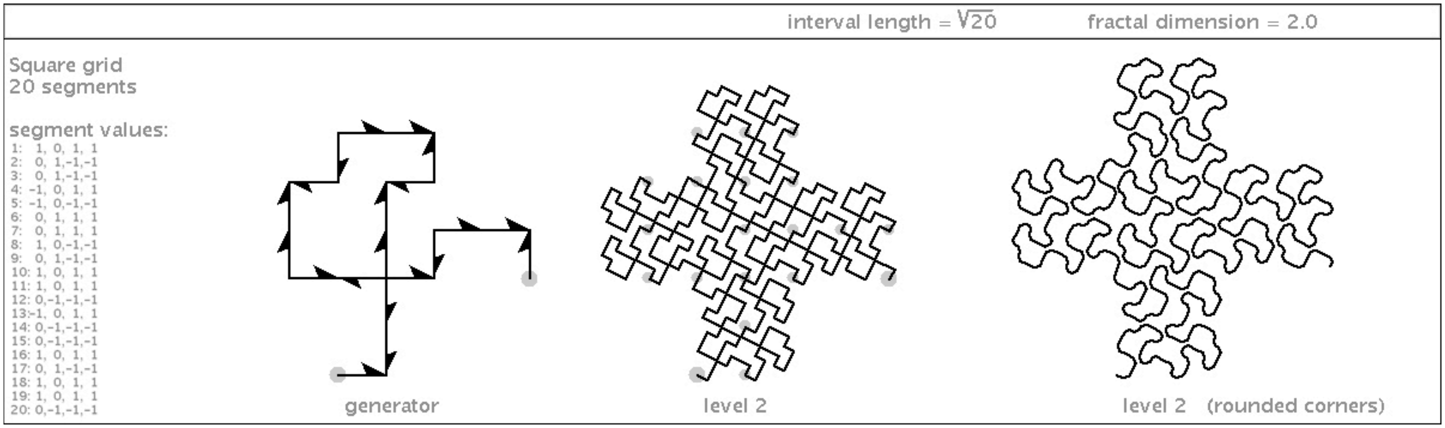
Remember the Anti-Gosper? Well, I suspect that the following curve has some things in common: it is roughly triangular and more tightly-packed than its hexagonal cousin.



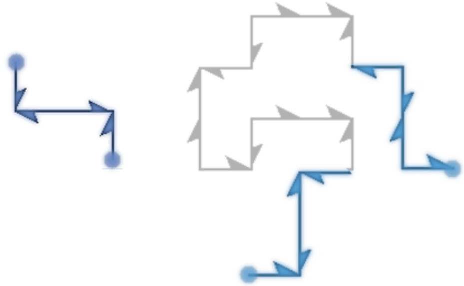
The √20 Square Grid Family

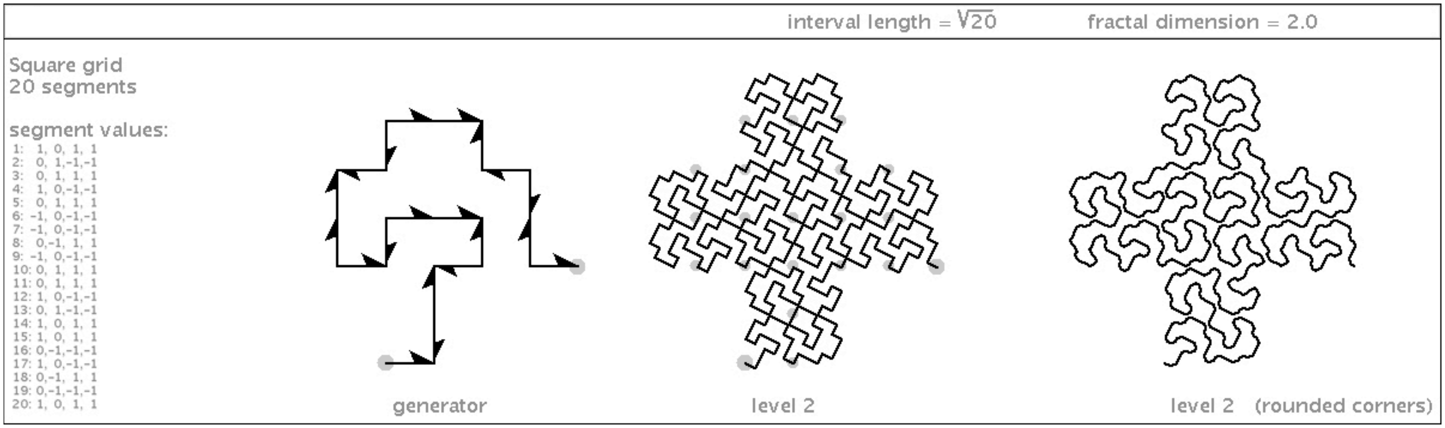
There are five fours in twenty. And so I wondered if I could build a space-filling curve by replacing the squares in Mandelbrot's Quartet with rotated copies of a √4 square grid family generator. Well, the Peano Sweep generator appears to do the trick:



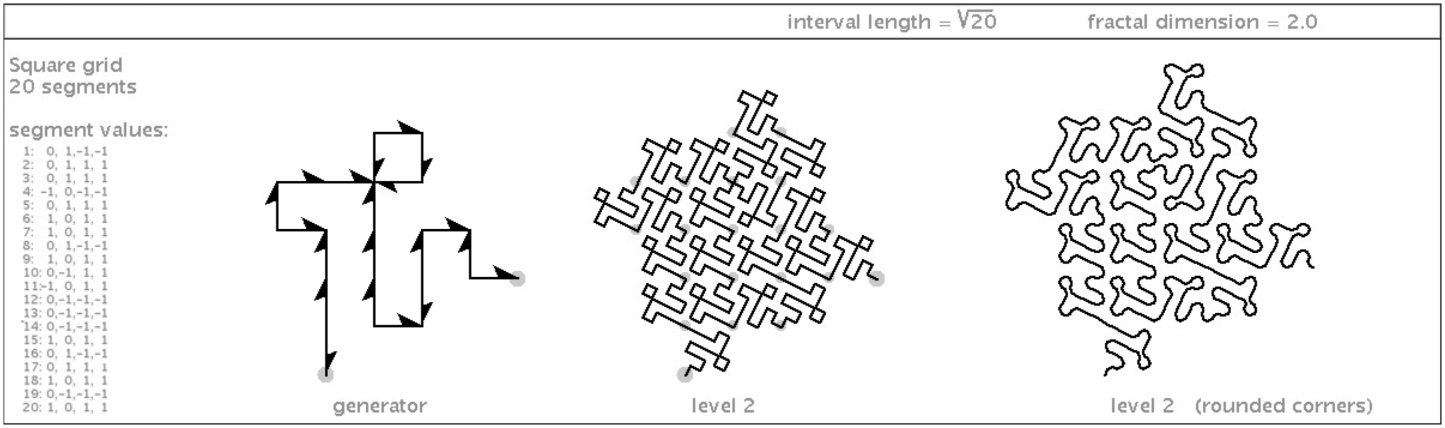


Here is a variation that uses a combination of the Peano Sweep generator and another four-segment shape that fractalizes to a square. It is a shape that would not normally stand on its own as a plane-filling curve generator. But in the context of the whole arrangement, it works perfectly.

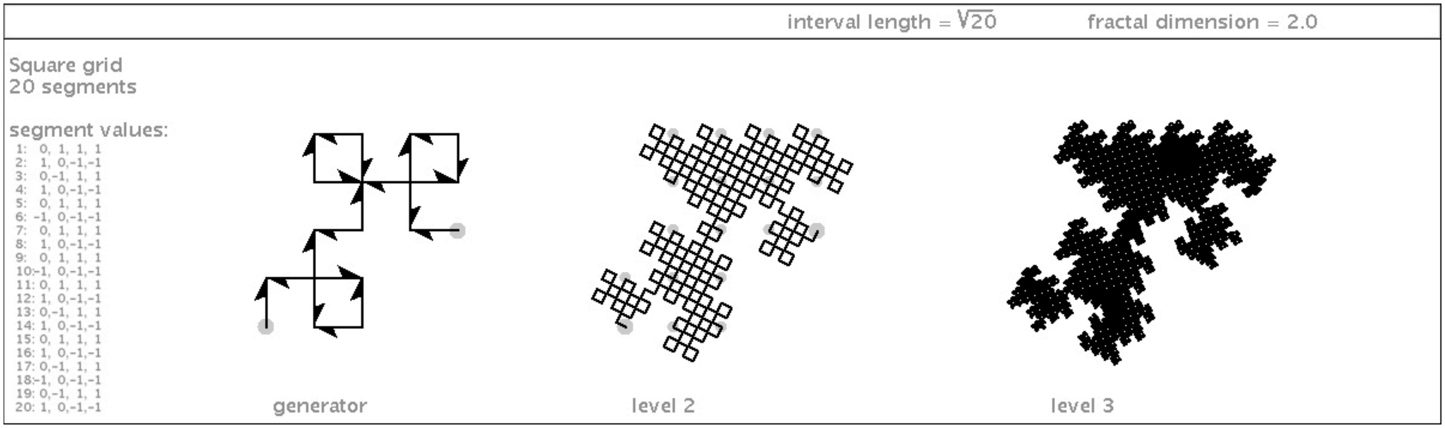




The following generator has a knotted-off square, like the √17 specimen we saw earlier.

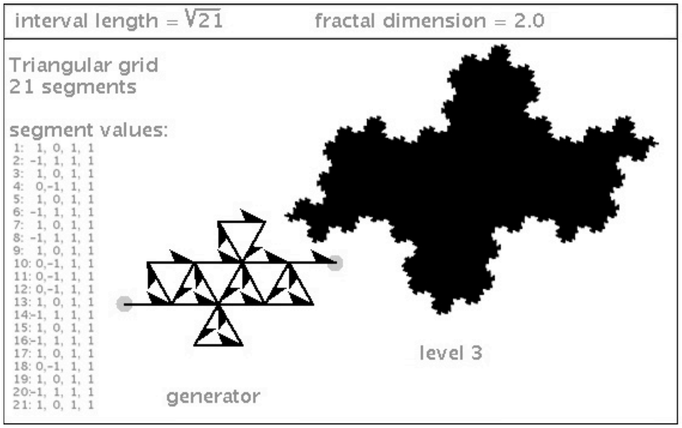


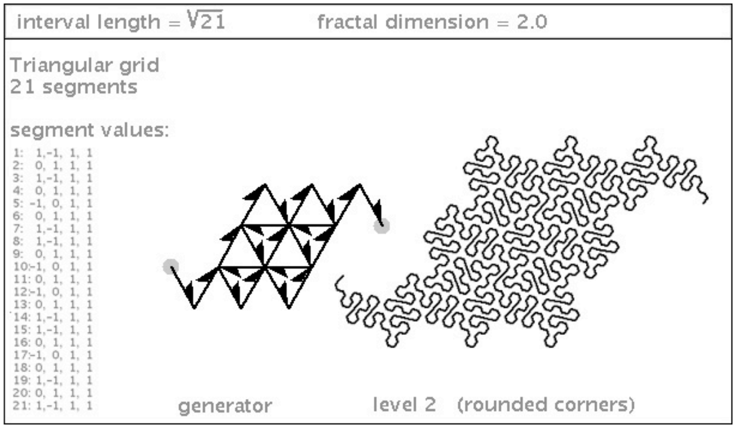
This curve is rather curious. It is a gridfiller with a bit of asymmetry.

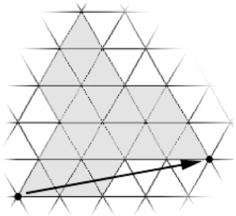


The √21 Triangle Grid Family

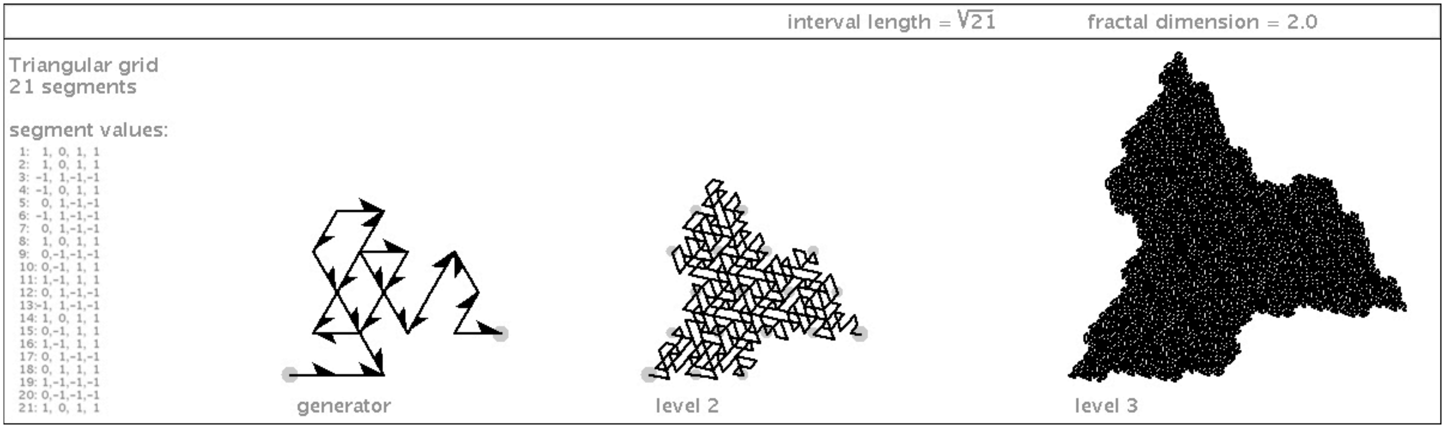
Of the √21 triangle grid family, I have found a few fat palindrome dragons. Two of them are shown below.

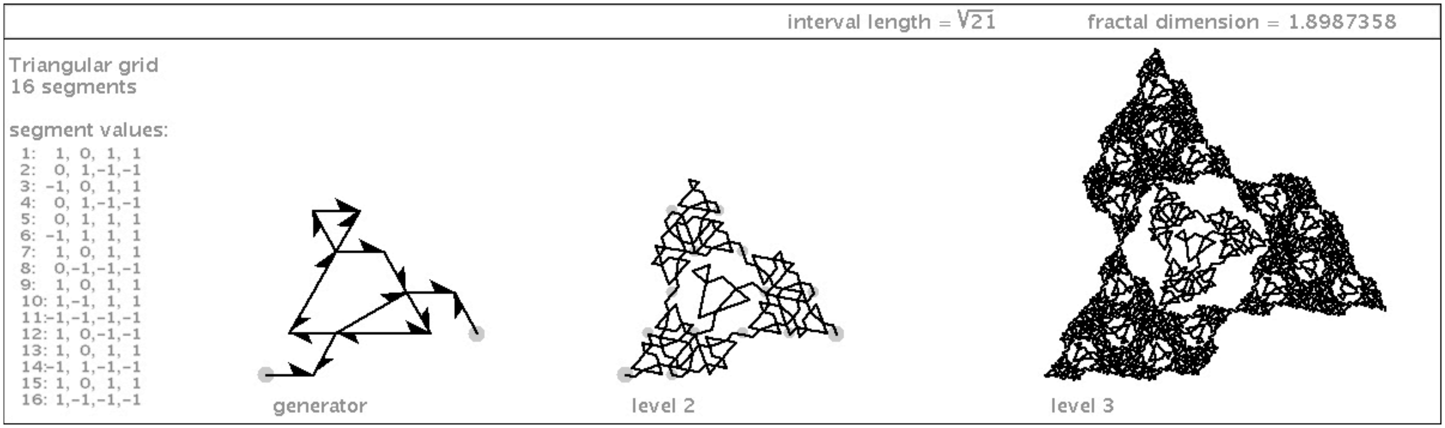


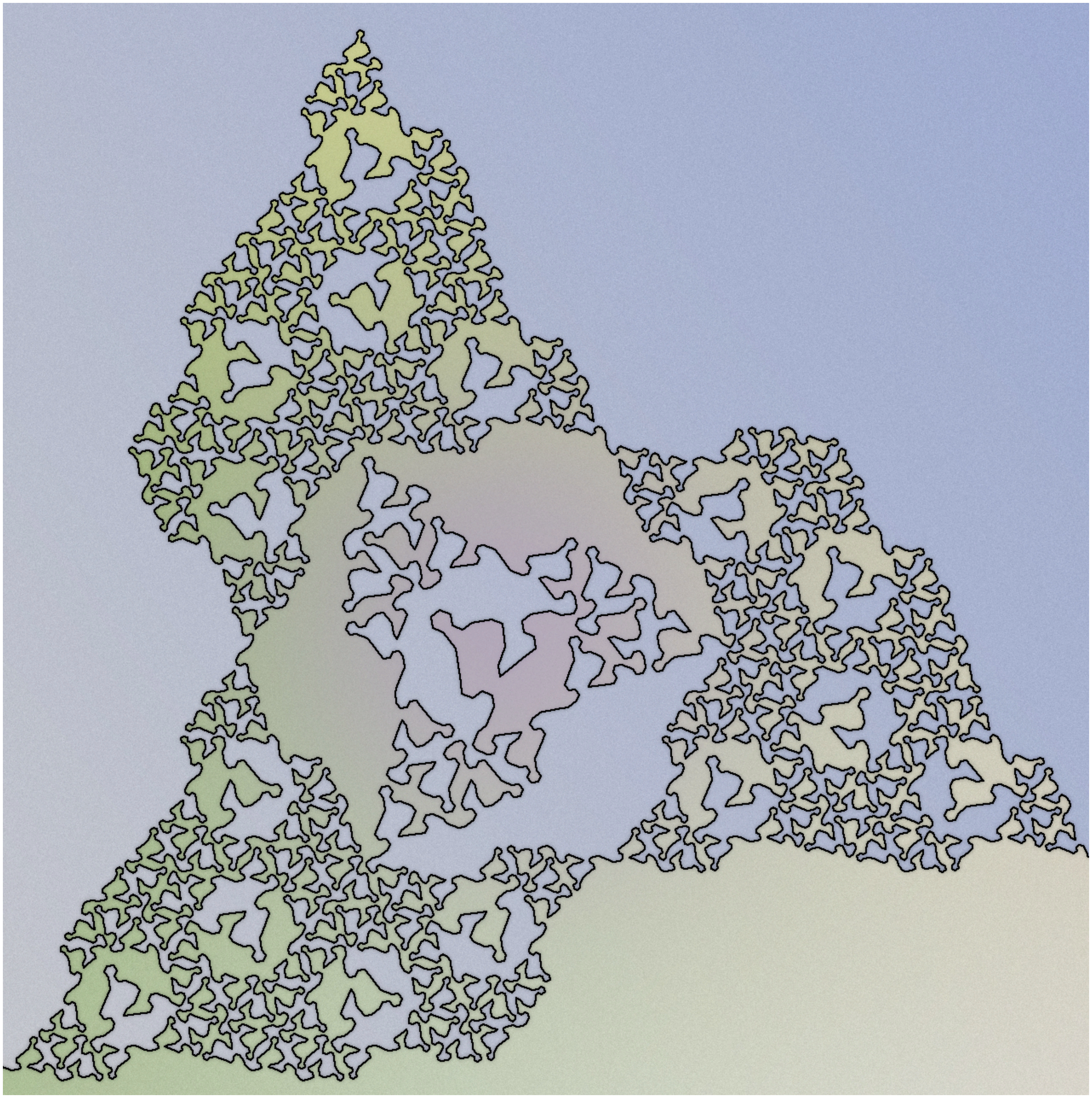




Since 21 is a triangular number, we can arrange the pertiles in a roughly triangular array. Below is a triangular specimen of this family, followed by a close cousin with dimension ~1.89. On the next page is a color rendering with rounded corners.

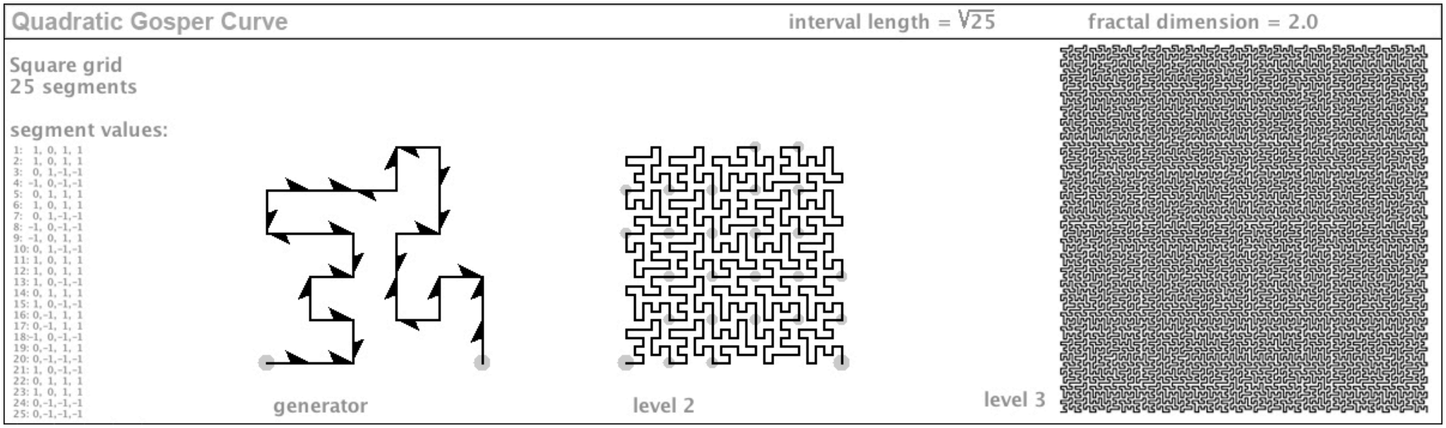






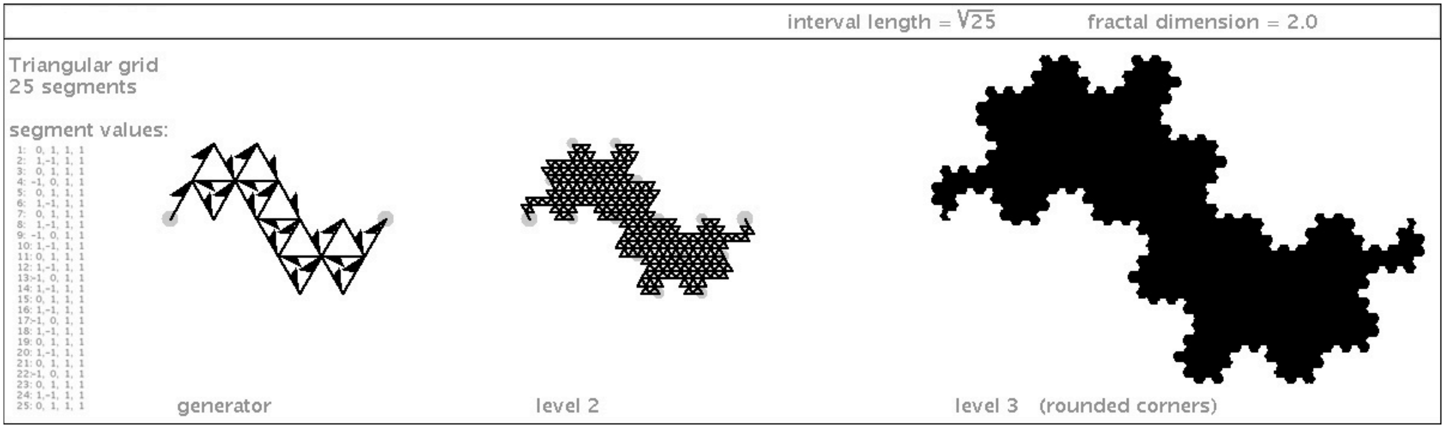
The √25 Square Grid Family

Now we come to another square number: 25. Of the √25 square grid family, I have only one specimen to show: the *Quadratic Gosper Curve*. It is attributed to F.M Dekking [5], and also Doug McKenna [19]. This is a self-avoiding curve that fills a square.



The √25 Triangle Grid Family

The √25 triangle grid family has a whale of a bumpy dragon, shown below.

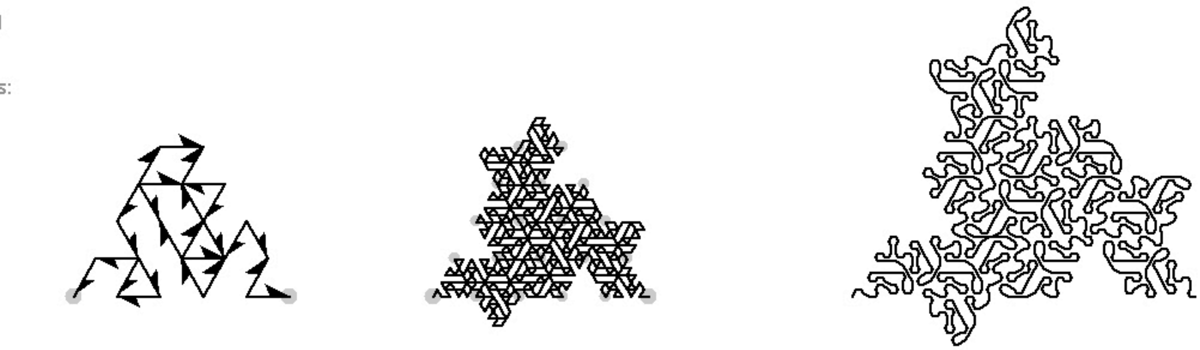


On the next page are three more specimens of this family. The third one is shown on the following page enlarged at level 3 with rounded corners.

interval length = $\sqrt{25}$ fractal dimension = 2.0

Triangular grid
25 segments

segment values:
1: 0, 1,-1,-1
2: 1, 0,-1,-1
3: 0,-1, 1, 1
4: 1, 0,-1,-1
5: -1, 1,-1,-1
6: -1, 1,-1,-1
7: 0, 1,-1,-1
8: 0, 1, 1, 1
9: 1, 0, 1, 1
10: 0,-1,-1,-1
11:-1, 0,-1,-1
12: 1,-1, 1, 1
13: 1,-1,-1,-1
14: 1,-1,-1,-1
15: 0, 1, 1, 1
16:-1, 0,-1,-1
17: 0, 1, 1, 1
18:-1, 1, 1, 1
19: 1, 0,-1,-1
20: 0,-1, 1, 1
21: 1,-1, 1, 1
22: 0, 1,-1,-1
23: 1,-1, 1, 1
24: 0,-1,-1,-1
25: 1, 0, 1, 1

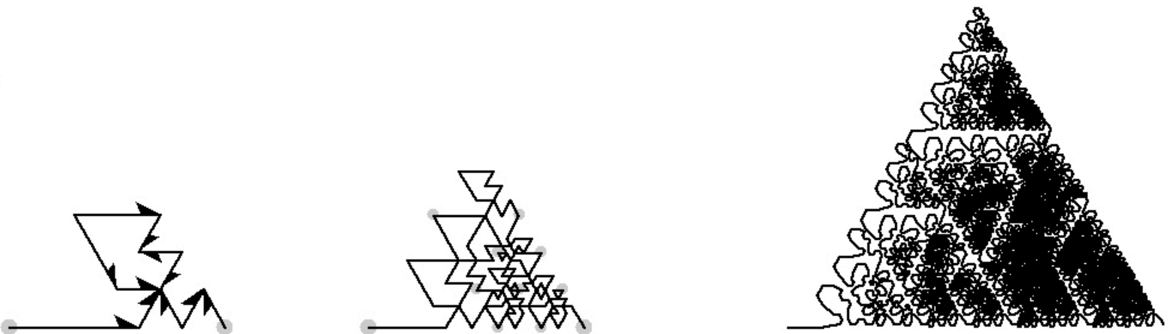


generator level 2 level 2 (rounded corners)

interval length = $\sqrt{25}$ fractal dimension = 2.0

Triangular grid
11 segments

segment values:
1: 3, 0, 1, 1
2: 0, 1, 1, 1
3: -1, 0,-1,-1
4: -2, 2,-1,-1
5: 2, 0, 1, 1
6: 0,-1, 1, 1
7: 1, 0,-1,-1
8: 0,-1, 1, 1
9: 1,-1,-1,-1
10: 0, 1, 1, 1
11: 1,-1,-1,-1

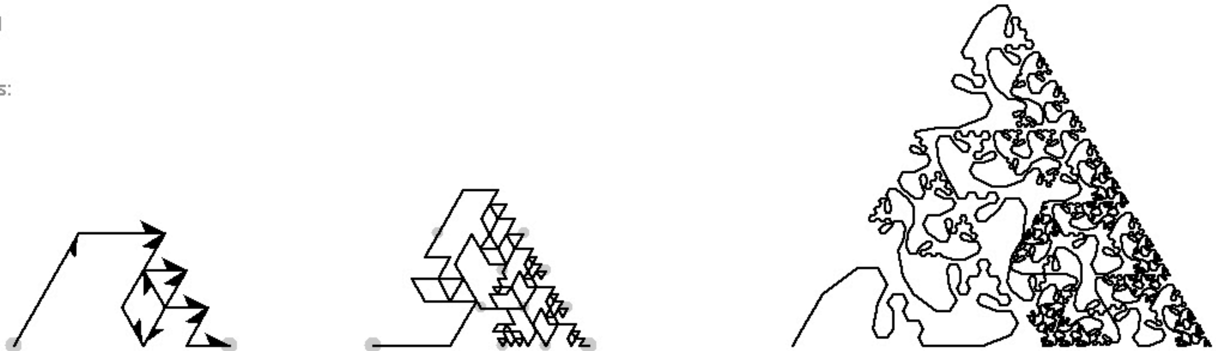


generator level 2 level 4 (rounded corners)

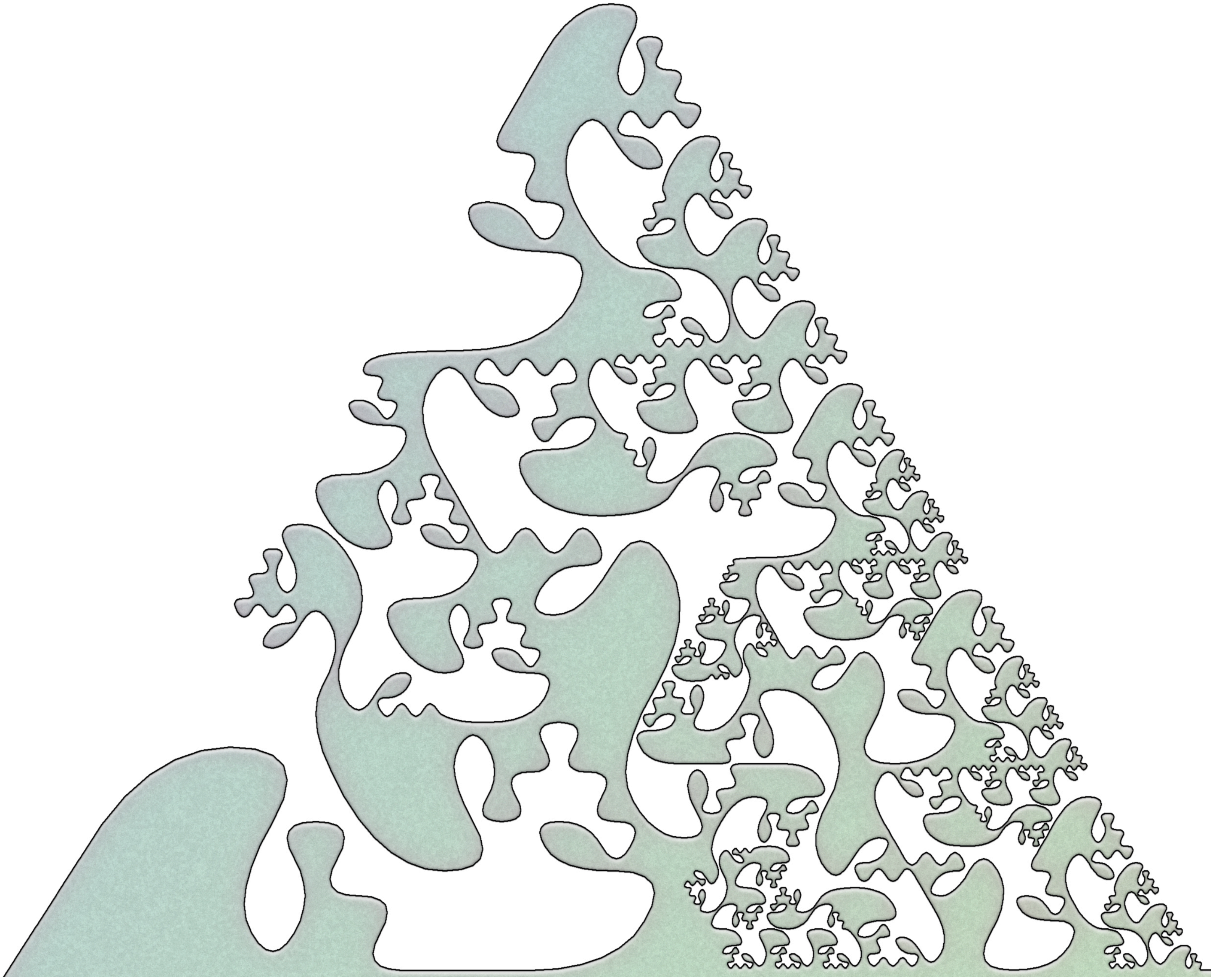
interval length = $\sqrt{25}$ fractal dimension = 2.0

Triangular grid
11 segments

segment values:
1: 0, 3, 1,-1
2: 2, 0, 1, 1
3: 0,-2,-1,-1
4: 1,-1, 1, 1
5: 0, 1,-1,-1
6: -1, 1, 1, 1
7: 1, 0, 1, 1
8: 0,-1,-1,-1
9: 1, 0, 1, 1
10: 0,-1,-1,-1
11: 1, 0, 1, 1

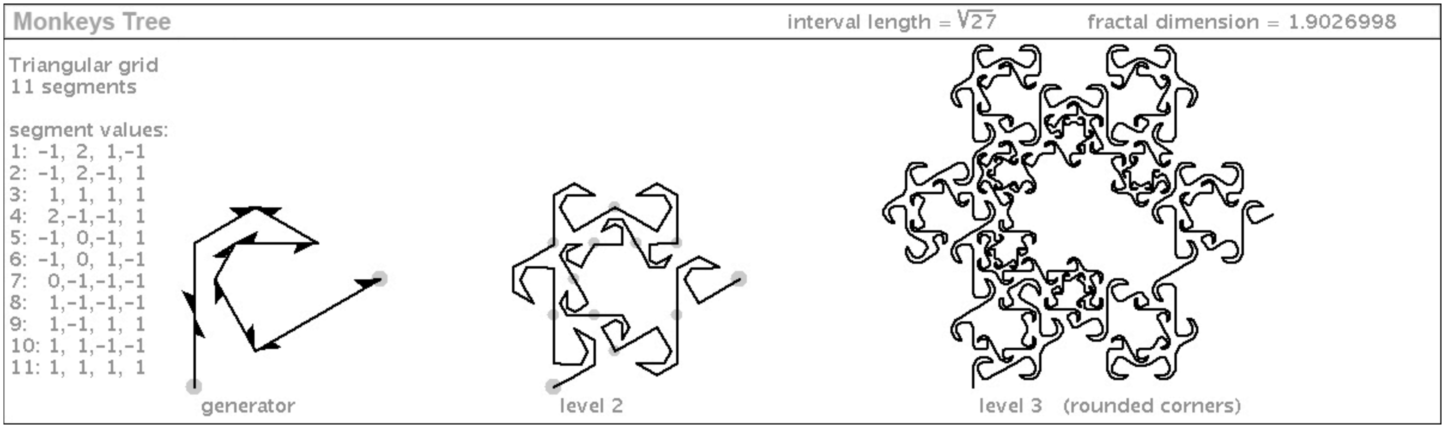


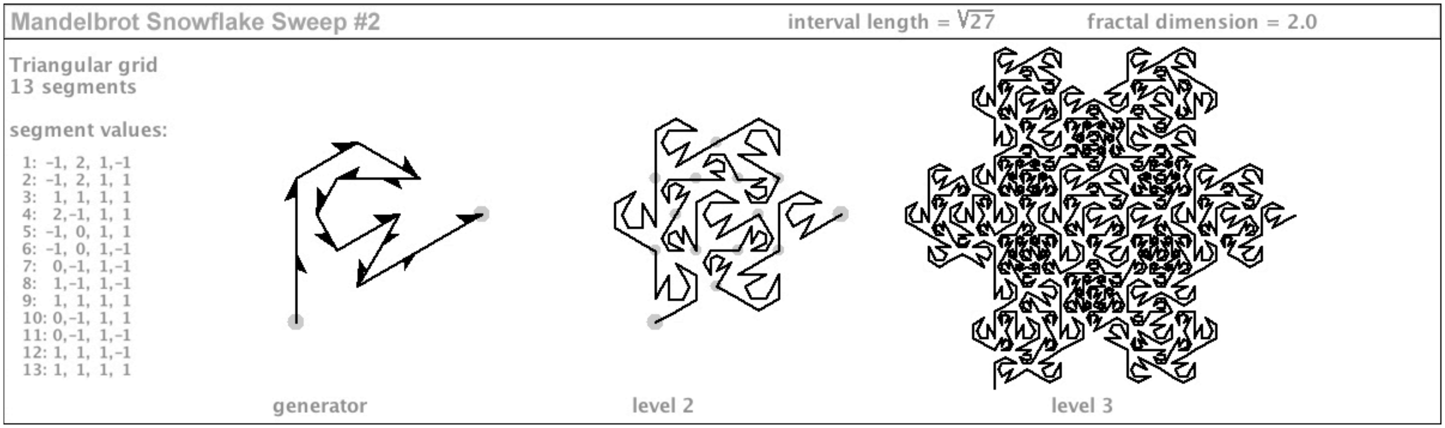
generator level 2 level 3 (rounded corners)

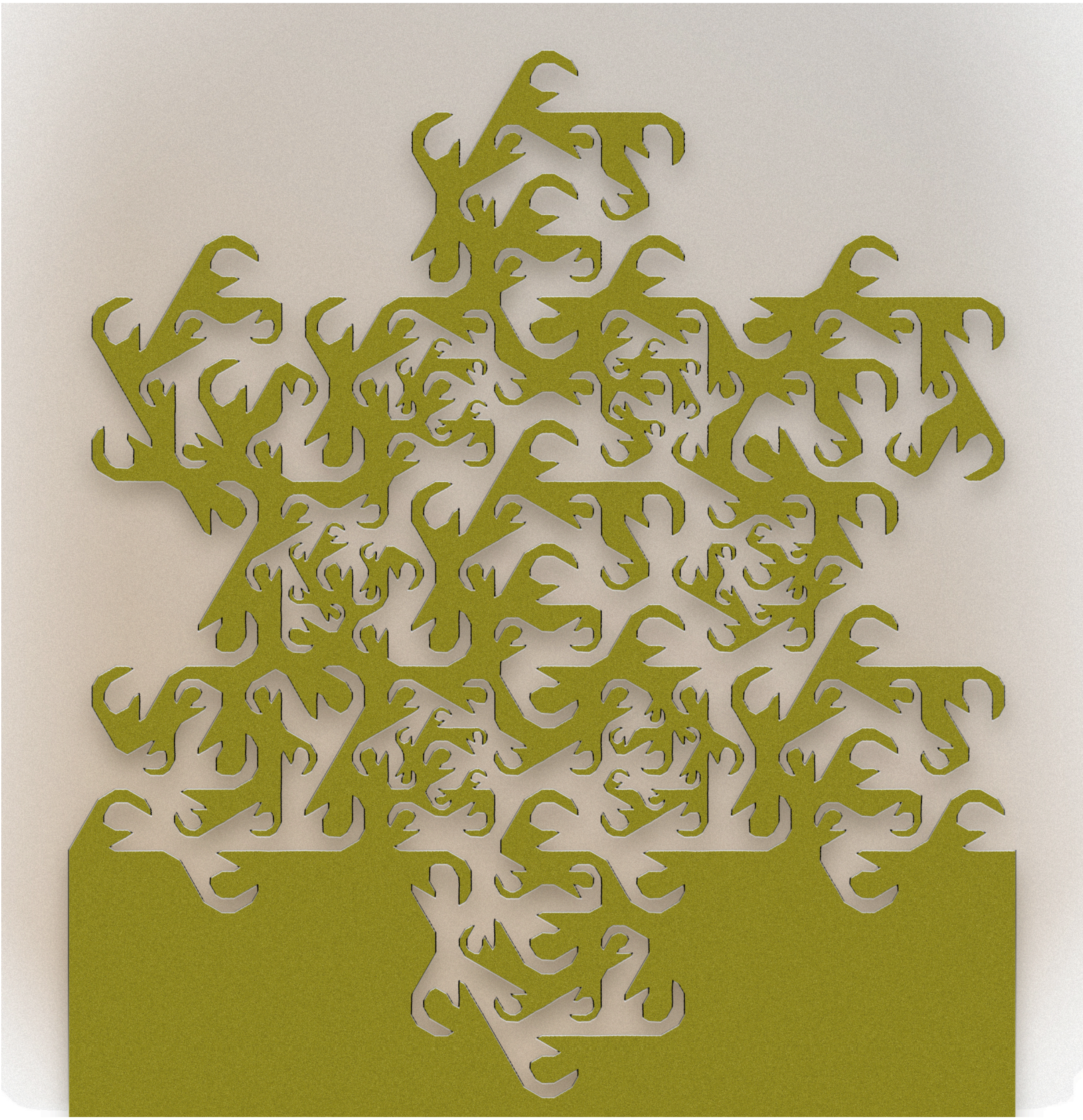


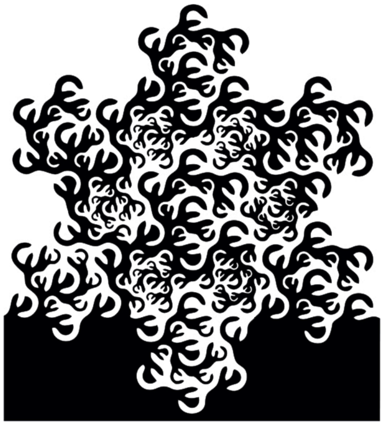
The √27 Triangle Grid Family

I want to show you two related specimens of the √27 triangle grid family. They were introduced in Mandelbrot's book, and they are variations of the Snowflake Sweep. Now, as we saw earlier, the snowflake sweep is a member of the √9 family, so why are these specimens √27? The reason is due to the nature of my taxonomy scheme, which requires that all generator segments lie between grid points. With a slight variation to my scheme, these curves could be represented in the √9 family, by relaxing this requirement. The smallest segments occurring within the interior of the generator would fall between the cracks, as it were. We can say that these specimens are closely related to the √9 family, except that they have a few small (and clever) genetic mutations that permit extra details to emerge in the crevices. The first specimen is one Mandelbrot called "Monkeys Tree". The second is a variant of the Snowflake Sweep, in which the whole generator is transformed and takes the place of the fifth segment. This is rendered stylistically on the next page.



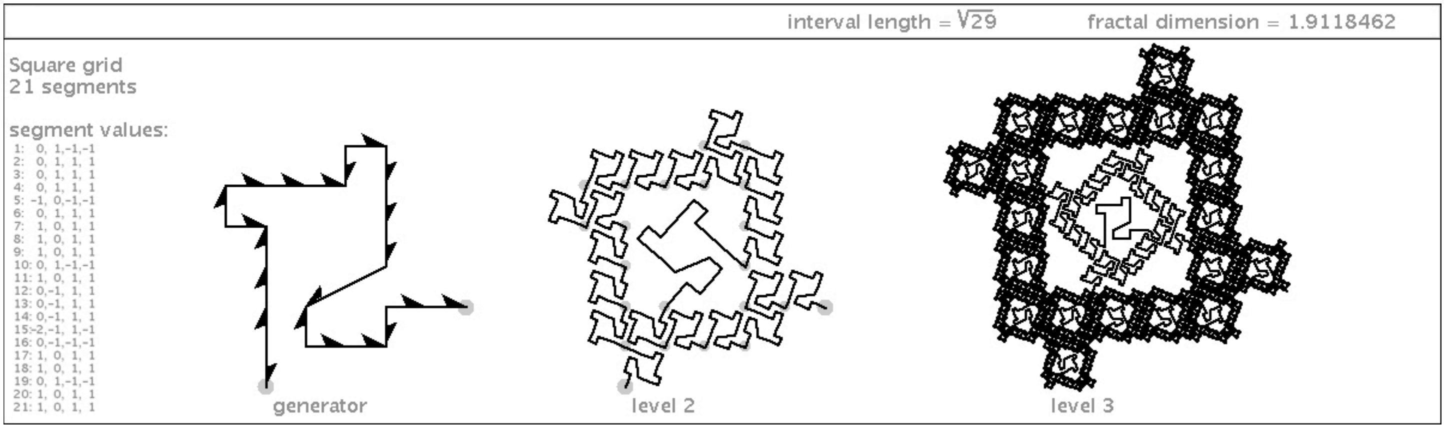


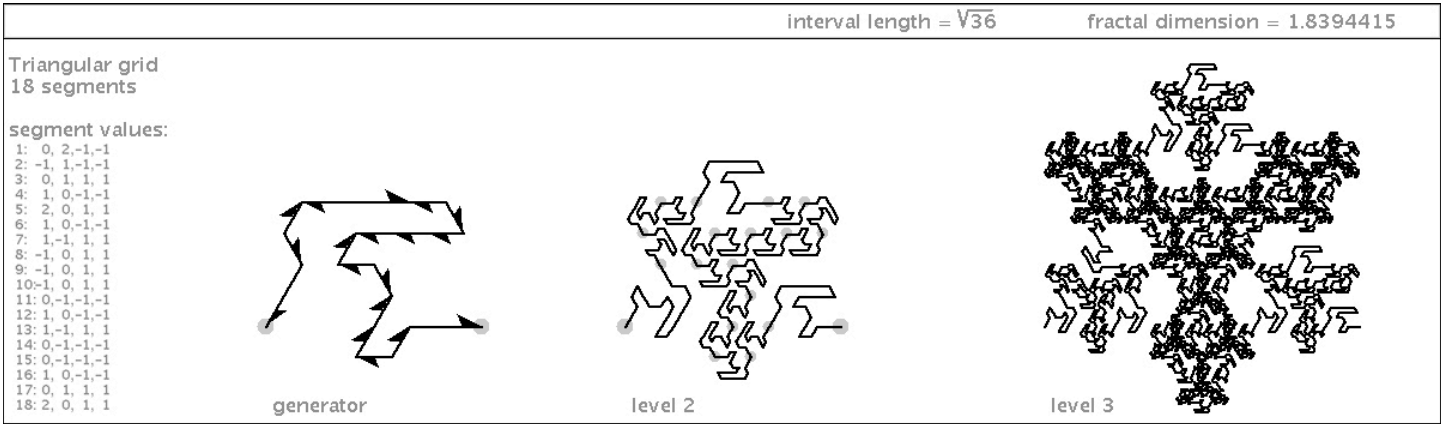


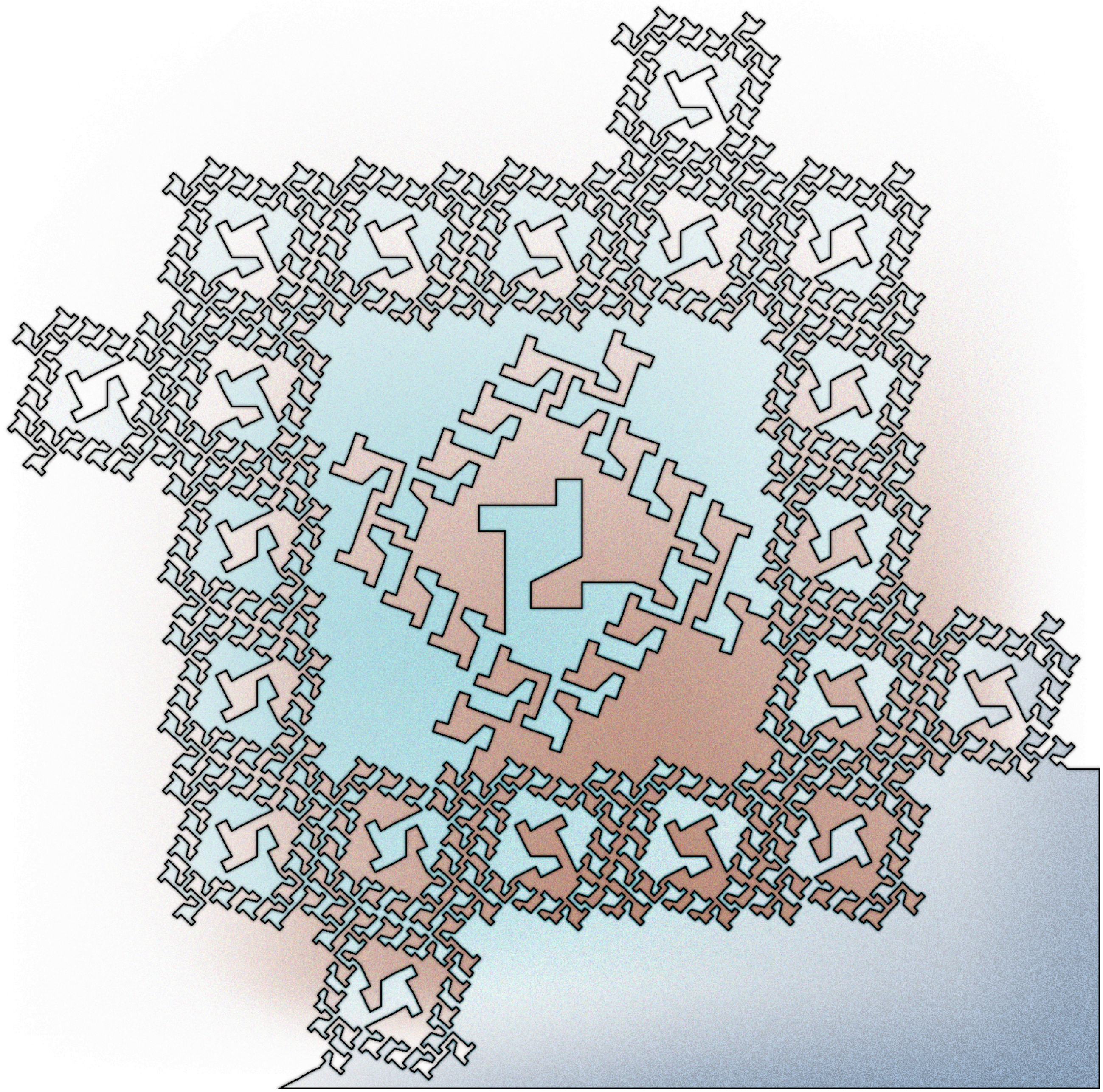


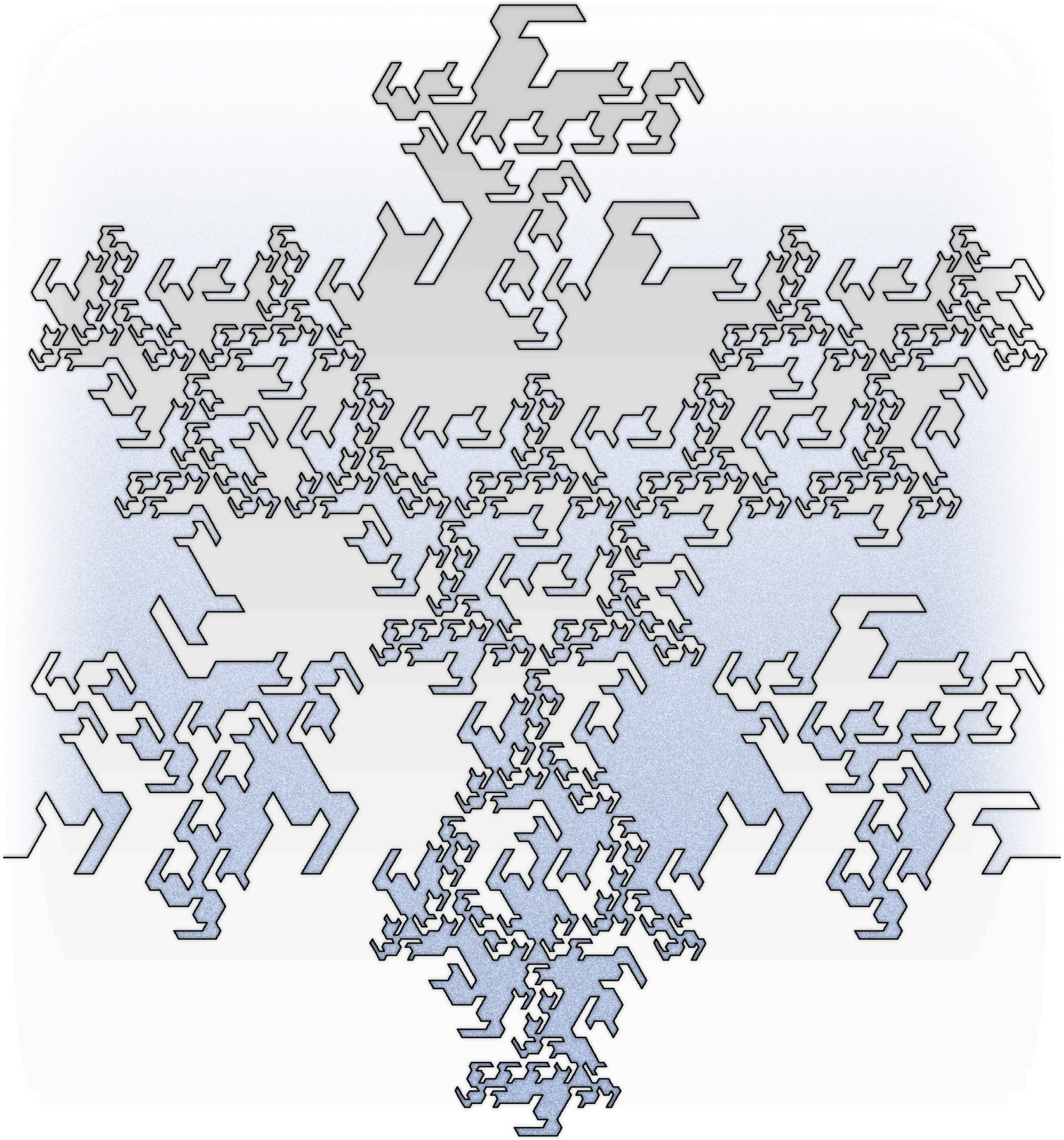
This specimen was rendered with loving curves and printed on the hardback cover of one of the editions of Mandelbrot's book. It is reproduced at left.

The two final specimens I want to show you are of the √29 triangle grid and √36 square grid families. These both have dimensions less than 2, but they are self-avoiding, and so they do not require rounded corners. Their diagrams are shown below, and on the next two pages are color renderings. The second one is my attempt at filling the Koch Snowflake. Enjoy the last two specimens of this book!





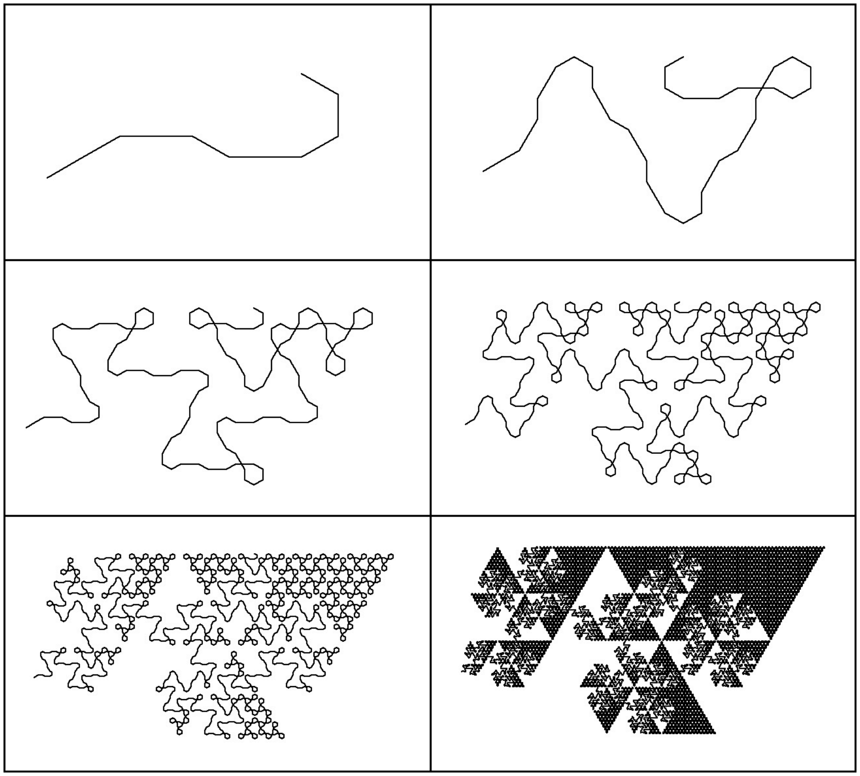




My Brain Fillith Over

I think that's enough brain-filling for one book.

I hope you enjoyed going on this expedition of fractal specimens with me. I have enjoyed discovering them, and I thoroughly enjoyed building this book. But let's not stop here. There are more fractal curves to discover, and there are more ideas, inventions, artworks, and insights about nature, the human mind, and the vast potential of mathematics – using the computer as an extension of the human eye and brain. For further adventures, visit me at fractalcurves.com where I will be expanding this Fractal Family Tree.



References

[0] Abelson, H., and diSessa, A. Turtle Geometry. MIT Press, 1986.

[1] Bandt, C. and Gummelt, P. Fractal Penrose Tilings I. Construction and matching rules. Aequationes Mathematicae, vol. 53, Numbers 1-2, Feb. 1997

[2] Barnsley, M. Fractals Everywhere. Academic Press, 1993.

[3] Carbajo, V. Fractal artworks published online at: http://www.carbajo.net/

[4] Fukuda, M. Shimizu and G. Nakamura, New Gosper Space Filling Curves, Proceedings of the International Conference on Computer Graphics and Imaging (CGIM2001) 34--38 . 2001

[5] Dekking, F. M. Recurrent Sets: a Fractal Formalism. Delft University of Technology, 1982.

[6] Fiedorowicz, Z. . √4 triangle grid family triangle curve variation, published at: http://www.math.osu.edu/~fiedorow/math655/examples2.html

[7] Gardner, M. In which “monster curves” force redefinition of the word “Curve”. *Scientific American*. 235. 1976

[8] Gelbrich, G. Fractal Penrose Tiles II: Tiles with fractal boundary as duals of penrose tiles. Aequationes Mathematicae, vol. 54, Numbers 1-2, Aug. 1997

[9] Hilbert, D. "Über die stetige Abbildung einer Linie auf ein Flächenstück", Math. Ann., 1891 (38), pp. 459-460.

[10] Hutchinson, J. Fractals and Self-Similarity. Indiana University Math. J. 30 (5) 713-747 1981.

[11] Karzes, T. Tiling Fractal curves published online at: http://karzes.best.vwh.net/

[12] Koch, H. "Sur une courbe continue sans tangente, obtenue par une construction géométrique élémentaire." Archiv för Matemat., Astron. och Fys. 1, 681-702, 1904

[13] Koch, H. "Une méthode géométrique élémentaire pour l'étude de certaines questions de la théorie des courbes planes", Acta Mathematica, Stockholm, 1906 (30), pp. 145-174.

[14] Mandelbrot, B. The Fractal Geometry of Nature. W. H. Freeman and Company. 1977

[15] Mandelbrot, B. The Fractal Geometry of Nature, 1977 (illustrating the quadric Koch Island)

[16] Mandelbrot, B. The Fractal Geometry of Nature, 1977 (describing the “Quartet”)

[17] McKenna, D. Fractal artworks, available at: http://www.mathemaesthetics.com/MathArtPrints.html

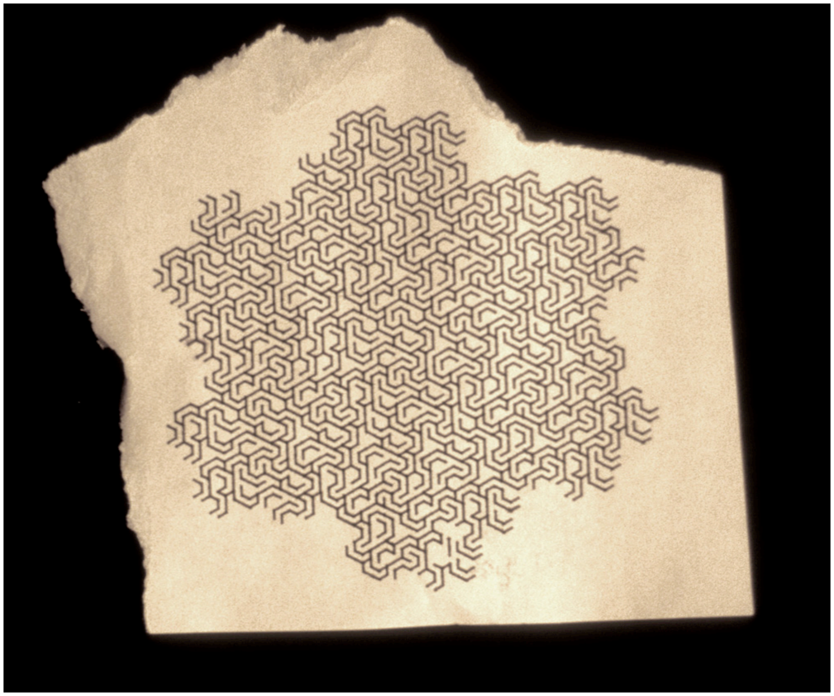
[18] Peano, G. "Sur une courbe, qui remplit toute une aire plane", *Mathematische Annalen* **36** (1): 157–160. 1890

[19] Prusinkiewicz, P. and Lindenmayer, A. The Algorithmic Beauty of Plants. Springer, 1990

[20] Sagan, H. Space-Filling Curves, Springer-Verlag, New York, 1991

[21] Schraa, W. Range Fractal, published online at: http://wolter.home.xs4all.nl/index.html

[22] Teachout, G. Spacefilling curve designs featured in the web site: http://teachout1.net/village/



12

About the Author

Jeffrey Ventrella is an artist and a software programmer. He writes about artificial life, virtual worlds, computational art, and human-computer interfaces. He lives in the San Francisco Bay area.